AMRL-TR-65-43

AD617740

A GUIDE TO PREPARING INTRINSICALLY PROGRAMMED INSTRUCTIONAL MATERIALS

R. E. WALTHER, Ed. D.
NORMAN CROWDER

U. S. INDUSTRIES, INCORPORATED

APRIL 1965

BEHAVIORAL SCIENCES LABORATORY
AEROSPACE MEDICAL RESEARCH LABORATORIES
AEROSPACE MEDICAL DIVISION
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO

A GUIDE TO PREPARING INTRINSICALLY PROGRAMMED INSTRUCTIONAL MATERIALS

R. E. WALTHER, Ed. D.
NORMAN CROWDER

FOREWORD

This report was prepared by the Educational Science Division of U.S. Industries, Inc., Silver Spring, Maryland, as a portion of the work required under Contract AF33(616)-6983. The actual writing was accomplished during the period 1 June through 31 October 1963, although the report reflects experience gained during previous portions of the contract dating from 15 December 1959. Dr. R. E. Walther and Mr. Norman Crowder, Principal Investigator for this contract, are the authors. The contract is in support of Task 171007, "Automated Training and Programmed Instruction," of Project 1710, "Training, Personnel and Psychological Stress Aspects of Bioastronautics." Dr. Theodore E. Cotterman was contract technical monitor for the Training Research Division of the Behavioral Sciences Laboratory during the preparation of this report.

This is one of a series of reports resulting from extensive research in automated technical training conducted by the Behavioral Sciences Laboratory, Aerospace Medical Research Laboratories, at Keesler Air Force Base, Mississippi, with the participation and support of the 3380th Technical School, Air Training Command. The general purpose of the project was the evaluation of programmed learning methods. The first report, MRL-TDR-62-78, described the early comparisons of the test performance of students using intrinsic programs with the performance of students given conventional instruction. The second report, MRL-TDR-62-79, compared the costs of instruction by intrinsic programs with the costs of conventional instruction. This report details a method by which effective intrinsic programs can be economically produced.

The authors acknowledge the contributions of the members of the Educational Science Division staff. In particular, they wish to give credit to Mr. Edgar Orloff, formerly Chief Editor, whose writing staff has shown tremendous ingenuity and flexibility in developing and trying new ideas, and to Mr. Frank Griffin, whose efficient operation of the Production Department has resulted in many of the ideas in this report for maintaining high quality at minimum cost. Many other individual employees, both present and past, contributed ideas and information to the content of this report.

This technical report has been reviewed and is approved.

WALTER F. GRETHER, PhD
Technical Director
Behavioral Sciences Laboratory

ABSTRACT

To aid those responsible for the preparation of intrinsically programmed instructional materials, the procedures and techniques developed by the Educational Science Division of U. S. Industries, Inc., have here been organized into a practical working guide. The organization of this report closely follows the sequence of steps required to produce an effective intrinsic program. Although other systems of programming are identified, this guide is specifically intended for use in the preparation of intrinsic programs in either book or TutorFilm format.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Philosophy of Auto-Instruction

Auto-instruction is automatic self-teaching. It differs from most "home study," "correspondence," and other forms of self-study in three important ways: (1) the material is presented via a "program"; (2) some method of continuous and immediate evaluation of the student's progress is provided; and (3) the instructional material (program) is presented in a device or format that can control the student's access to and through the material and that makes commitment mandatory on the part of the student. To be auto-instructional the system must meet all three of these criteria without the aid of a live instructor. This report describes one system that meets these requirements.

The term "program" is used to designate the body of knowledge or information as it is organized into a series of sequential concepts for self-instructional purposes. There are several approaches that dictate the format and organization of the concepts. The two major approaches are called intrinsic programming and linear programming. A linear program has a single, predetermined sequence of steps. Error responses are not immediately corrected or immediately repeated. The approach attempts to avoid error responses and to shape behavior according to a predetermined sequence.

This report describes a method of preparing a program which uses the intrinsic approach. An intrinsic program is a body of knowledge or information organized into a series of sequential concepts, each followed by appropriate multiple-choice questions used to evaluate and insure the student's understanding of the material. The multiple-choice questions are specified in this format simply because they limit the universe of possible responses to a manageable number of typical errors that might logically be made.

By limiting the universe of possible responses through the use of multiple-choice questions, we provide a practical method for continuous and immediate evaluation of the student's progress. This is possible because the student's choice of an answer to a multiple-choice question can be used diagnostically to direct him to new material, and the student who chooses one alternative can be directed to different material from that to which a student choosing a different alternative is directed. Thus, if an alternative is representative of a logical type of error, the person writing the program can prepare corrective expository material appropriate to that response and to the needs of the student selecting that response. This fulfills our second requirement for auto-instruction.

The third requirement for auto-instruction involves a device or format that can control the student's access to and through the material in such a way that the student must "commit himself" in order to proceed. Commitment is an intense form of personal involvement that precludes rationalization or self-deceptive alibis. In some forms of programming it is possible for the student to "peek ahead" at the response confirmation before he has really solved the problem or made a decision. This preview of the correct response then influences the student's decision or answer without necessarily requiring his active participation.

In a system that can control commitment by preventing "peeking ahead" the individual has some degree of prestige at stake when he commits himself to the selection of one of the alternatives. This involvement may be minimal at the beginning of the program but early successes tend to raise the student's aspiration for continued success.

The extent of commitment is further intensified by occasional failures. The failures, if not too frequent, complement and personalize the successes. Success is a rewarding experience only when it is achieved in the face of possible failure. Unless a string of successes is interspersed with occasional failures, the successes lose their significance and the student loses his involvement. By setting up a random success/failure pattern with successes materially outnumbering the failures, a high degree of involvement or "commitment" can be achieved.

Brief Explanation of Intrinsic Programming by Norman Crowder

The pupil-tutor relationship was the model for the auto-instructional technique known as intrinsic programming. The characteristic feature of the pupil-tutor relationship is interaction. The pupil responds to what the tutor does, and the tutor responds to what the pupil does. The major structural features of intrinsically programmed material are designed to permit this same sort of interaction without a live tutor; the rationale of the method derives from the fact that the necessary two-way responsiveness can be achieved with straightforward, practical devices.

In the simplest "scrambled book" or TutorText® format the student is given a short discussion of the material to be learned, followed by a multiple-choice question designed to test the point just discussed. Each answer alternative has a page number beside it. The student chooses what he believes is the correct answer to the question and turns to the page number given for that answer. If he has chosen correctly, the page to which he thereby comes will contain the next unit of material to be learned and the next question, and so on. If he has chosen an incorrect answer, the page to which he thereby comes will contain a discussion of why the answer chosen is incorrect, and, following this discussion, an instruction to return to the original question page to try again. He will not come to the next unit of new material until he has chosen the correct answer, of course, although in choosing incorrect answers he will come upon new discussion of the old material.

The pages in a TutorText are randomly arranged: that is, the page numbers given with the answer choices are not consecutive or in any other obvious order. Thus the student cannot ignore the question and routinely pass to the next page of material; he must commit himself to one of the answer choices, or else choose blindly, but he cannot ignore the question and pass to the "next page" of instruction, since the "next page" is not the sequentially numbered next page, but the page whose number is given with the right answer choice.

The basic intrinsic programming technique, then, amounts to nothing more than the inclusion of multiple-choice questions in relatively conventional expository text and the use of these questions to check continually on the student's progress through the material and to furnish specific remedial material as it is required. In intrinsic programming the questions serve a primarily diagnostic purpose, and the basis of the technique is the fact that the diagnosis so made can be promptly utilized to furnish specific remedial material to the student.

The inclusion of the question and its answer choices in each page of expository text brings about changes in both text style and format that the skillful program writer will use for a variety of auxiliary purposes. Thus, the question may be used to draw the student's attention sharply to the key point of a paragraph; solutions to problems may be suggested by way of the answer choices provided with the questions; answering the question may provide the student with useful practice with the concept involved, and so forth. The material from which the student is to learn includes the expository text, the question, the answer choices provided, and the remedial material provided for each of the wrong answers. In other words, one does not prepare an intrinsic program merely by chopping up expository text into paragraphs and providing a multiple-choice question for each paragraph. However, the fact that a skillful writer will make the structural features of the format serve useful auxiliary ends, just as any competent craftsman will get the most from a technique, should not divert attention from the fact that the primary purpose of including the questions is diagnostic, the diagnosis being desired in order that prompt remedial action may automatically be taken.

The basic structure of intrinsically programmed material is quite simple. In each program step, the student is given a "unit" of material to read, usually a paragraph of thirty to seventy words. This material is followed by a multiple-choice question. The student's answer choice determines directly and automatically what material he will see next. If he chooses the right answer to the question, he is automatically presented with the next paragraph of material and the next question. If he chooses an incorrect answer, he is automatically presented with material written specifically to correct the particular error he has just made. At the end of this correctional material the student will, in the simplest case, be directed to return to the original presentation to have a second try at the original question, having completed a first order branch. However, the material at which the student arrives by making an error may be the start of a "Sub-program," or sub-sequence, of instructional material and questions in which the originally troublesome point is explained in smaller steps or with a different approach. Such an arrangement of material would be an example of second order branching.

The student works through a sub-sequence just as he does through the main program, advancing when he chooses correct answers to the questions he encounters, coming to specific remedial material if he makes errors. Sub-sequences of any complexity desired can be prepared. When the student has worked through the subsequence, he may be returned to the point in the main sequence at which he made his initial error, to a previous step in the main sequence, or to a succeeding step in the main sequence. This is arranged at the option of the programmer and then takes place quite automatically, as a function of the answers that the student chooses.

The crucial and identifying feature of intrinsically programmed materials is the fact that the material presented to each student is continuously and directly controlled by the student's performance in answering questions. To permit this step-by-step control of the program by the student, the questions are put in multiple-choice form. The choice of an answer to a multiple-choice question can be directly translated into a distinct physical act (turning to a particular page or pushing a particular button on a machine) which can then bring the appropriate material into view.

Much of the criticism of programmed instruction proceeds from the belief that students studying programmed material are not required or encouraged to think out a problem consciously and deliberately, or given any opportunity to stretch their mental muscles in reaching for generalizations or conclusions. The criticism can be minimized by the intrinsic programming technique. Students studying intrinsically programmed materials can be challenged at any level we wish to set, since means are provided to assist those who do not meet the challenge successfully.

In summary, the technique of intrinsic programming postulates that the basic learning takes place by an unanalyzed process during the student's exposure to written, or, in general, symbolic material. The major function served by the multiple-choice questions is to test whether the postulated learning actually took place. The test result is used to control the next material the student sees, either advancing the student or supplying remedial material as indicated. The result is material that automatically adapts to individual differences among students and which allows us to set the difficulty level our educational objectives and subject matter require. It is also recognized that the inclusion of the questions serves other desirable purposes, such as keeping the student active, making it clear to him what he is expected to learn from the basic material, keeping him informed of his progress, and other motivational and practice purposes; but the basic purpose of the questions asked is to control the presentation of the material.

<u>Purpose, Limitations, and Organization of This Guide</u>

The purpose of this guide is to describe the techniques and procedures developed by the Educational Science Division of U. S. Industries, Inc., for the preparation of intrinsically programmed materials for use in auto-instructional systems. The procedures described in this guide were originally conceived and tested in the process of preparing instructional materials for use in evaluating the effectiveness of programmed learning under the present Air Force contract. The material is intended to guide inexperienced program producers through the initial phases of their programming efforts with a minimum number of false starts.

The instructions, suggestions, and techniques described apply only to the system known as intrinsic programming. Although an auto-instructional system may combine or incorporate other programming methods and formats, these techniques will not be dealt with in this report. The auto-instructional system on which this report is based was completely intrinsic in its format.

The organization of the material in this guide follows closely the sequential order by which the program itself must develop. This guide, like a typical program, begins by examining an established need. From this need a plan evolves so that costs can be estimated, progress charted, and records kept.

After the groundwork has been laid, personnel must be selected and trained in the use of the technology and the intrinsic program format. Next the program must be polished, tests and ancillary materials prepared, and the manuscript scrambled.

Preliminary student testing, revision and final polishing follow next in the sequence. Production and validation complete the cycle. It is this sequence that has determined the organizational pattern for the material in this guide.

LOGISTICS OF PROGRAMMING: PLANNING

The first step in programming is to determine that there is a training need. This need may result from technological changes, new assignment of responsibilities, or inadequacies in performance of duties by personnel.

Assuming that a training requirement has been established and that the subject is either not being taught or that the present method of teaching is not satisfactory, then it must be decided whether to use programmed instruction or some other method. Programmed instruction has unique advantages in certain situations and every effort should be made to capitalize upon them.

Programmed instruction should be considered in situations where there are insufficient numbers of skilled instructors available, because it provides a consistently high level of training with no variation in content. It is one of the most effective ways of guaranteeing that every trainee will receive precisely the same course content.

Programmed instruction should be considered as a preferred method in situations where the trainees are widely dispersed geographically or when, for other reasons, it is not convenient to assemble the trainees in instructional groups. Examples of such situations would be on-the-job training and orientation of personnel working staggered shifts or hours, or for instruction and training during periods when regular training is not offered.

Programmed instruction is more suitable for training that is relatively stable in content than for rapidly changing subjects. The revision of programs, like the revision of training films, is more difficult and expensive than the revision of live instruction. For this reason, care should be exercised not only in the selection of subjects for programming, but also in the preparation of the programs to exclude material that will either date the program or hasten its obsolescence.

At present, the most practical figure for use in determining the cost of programmed instruction is the cost per student hour. This is found by dividing the total cost of the program or the training by the total number of hours of instruction to be provided. For example, assume that there are ten thousand trainees per year who normally receive a one-hour orientation course. Assume that it costs $1,000 to develop this course in intrinsically programmed format. Dividing the $1,000 program costs by these 10,000 trainees gives a per-trainee cost of ten cents. This compares very favorably with the average cost of instruction in our public schools throughout the United States, which is 22 cents per student hour. * If per-student-hour training costs for other methods of instruction are available, this is a valuable method of appraising the economic feasibility of programmed instruction in a given situation. The programmed instruction costs are obtained from the project plan schedule, (discussed below).

Training by a live instructor is a relatively constant expense when calculated on a per-student basis. On the other hand, programmed instruction involves a relatively high initial expenditure but relatively low on-going expense. For this reason, the cost per student can be extremely low if the number of trainees is very large. When the number of students to be trained is very great, even a program subject to rapid obsolescence can provide high-quality training at low per-student cost. At the other extreme, a stable program used for training few students can be economical if the program is used for a sufficient length of time.

Another factor to be considered in making the decision about whether to use programmed instruction is the time available to develop and prepare the program. The most important factor in the success of good programs is the meticulous effort put into their preparation. Whereas a live instructor can put a course together rapidly and improve it as he goes along, programmed instruction must be carefully planned and highly polished before it is released.

To summarize, programmed instruction is generally more efficient or economical in the degree to which present methods are not satisfactory, whenever large numbers of trainees are involved, whenever a subject is basic and stable in its content, whenever there is an inadequate supply of trained instructors, whenever trainees are dispersed either geographically or in time, whenever there is adequate time for program preparation, whenever course content must be covered in a consistent fashion, and whenever a consistently high level of training effectiveness must be maintained. Additional comments on the implementation and utilization of programs may be found in Appendix A.

---

*School of Management, 1962 - 1963 Cost of Education Index, Page 107, January 1963.

The Project Plan and Schedule

Good programs provide effective instruction and economy in use, once developed. However, all programs are expensive to develop. It is therefore important to establish a realistic budget before undertaking a programming project. There are a number of approaches to the problem of estimating and obtaining accurate budget data for projects involving the preparation of programmed instruction. The approach described in this report was originally developed for use in producing materials in an earlier phase of the present Air Force contract work and has been subsequently revised and refined through continued use in other projects.

The procedure which makes use of the Project Plan and Schedule form (figure 1) has proven flexible enough to be useful in a wide variety of programming projects: simple to complex, short to long, intrinsic, linear, mathetical, and combinations of these. It includes enough detail to permit accurate cost estimating and yet is not too detailed to be practical. It provides a systematic way to structure what tends to be a nebulous problem.

The Project Plan and Schedule serves first to organize the data and information required to put a dollar and man-hour value on the project. Second, it serves as a guide to the individuals working on the project concerning budget and man-hour limitations in respect to their area of activity. Third, it provides a systematic method of accumulating historical data concerning projects, project directors, and individuals or groups working on phases of a project. However, the most important use of the Project Plan and Schedule is to assist management in making an intelligent decision concerning the feasibility of undertaking a given project.

The person preparing the initial Project Plan and Schedule should be thoroughly familiar with the preparation and production of programmed materials. If possible, this should usually be the individual who will become the Project Director if the project is approved. Since many decisions must be made on the basis of minimal information, it may be wise in some cases to secure a competent outside consultant to assist in the preparation of the Project Plan and Schedule.

Assuming that a training need has been established, that programmed instruction has been selected as the preferred training technique, and that it has been decided to prepare a Project Plan and Schedule and to cost the project; then the steps which follow are recommended to complete the suggested Project Plan and Schedule form shown as figure 1.

Project Number

The Project Number is an accurate way of identifying the project. If programming is a minor activity of the unit, the project number might well fall within the system used by the unit for identifying other similar projects. If the unit is primarily concerned with programming or expects to be increasingly involved in programming, then it may be helpful to establish a separate project number file for accumulating data relating only to programmed instructional projects.

Title of Project

In most cases, a short title or descriptive work title only will be given at the time the project plan is developed. This may or may not become the final title but will serve to identify the project in discussions.

Project Director

In many cases, the Project Director will not be known at the time the project plan is developed. In such cases, this space may be left blank or the name of the individual preparing the Project Plan and Schedule may be given.

Basic Objective

The basic objective is a simple clear statement of the intended purpose of the Project Plan and Schedule. It is not necessarily the same as the behavioral objective of the program that is being planned. For instance, the basic objective of the Project Plan and Schedule may be to secure data, to evaluate the economic feasibility of preparing the program. It may be to determine if the

Sheet 1 ________

PROJECT PLAN AND SCHEDULE

Project Number ______________________________ Date __________

Title of Project ______________________________

Project Director ______________________________

BASIC OBJECTIVE

DESCRIPTION OF WORK (whenever possible, treat outside labor as man-hours, not material):

1. Course Planning (Number of consultant man-hours? Materials?)

Total man-hours ______ Materials $ ______ Travel $ ______

2A. Writing and Editing—Text (Number of Pages? Which Sections, Lessons, Blocks, Parts?)

Total man-hours ______ Materials $ ______ Travel $ ______

2B. Writing and Editing—Test Sequences (Number of test sequences? How long? What Type?)

Total man-hours ______ Materials $ ______ Travel $ ______

Figure 1

Project Plan and Schedule

Sheet 2________

PROJECT PLAN AND SCHEDULE (continued)

3. Other Editorial Work/Technical Review (Number of consultant man-hours? Staff man-hours? Student man-hours? Materials?)

Total man-hours______ Materials $______ Travel $______

4. Scrambling and Scramble Checking (Number of pages of text? Test sequences? Simple or Complex?)

Total man-hours______ Materials $______

5A. Typing—Other than Mathematical (Number of pages?)

Total man-hours______ Materials $______

5B. Mathematical (Number of pages?)

Total man-hours______ Materials $______

5C. Manuscript Reproduction (Number of pages? Number of Copies? Type of Process?)

Total man-hours______ Materials $______ Travel $______

6A. Illustrating—Simple Line Art (Number of Drawings?)

Total man-hours______ Materials $______

Sheet 3________

PROJECT PLAN AND SCHEDULE (continued)

6B. Illustrating—Complex Line Art (Number of Drawings?)

Total
man-hours______Materials $ ______

6C. Illustrating- Other (Number of drawings? Description?)

Total
man-hours______Materials $______

6D. Splicing (Total number of Illustrations? Total pages of Manuscript?)

Total
man-hours______Materials $______

7A. Filming (Number of prints? Number of frames? Sales samples required?)

Total
man-hours______Materials $______Travel $_______

7B. Film or Galley Checking (Number of checkers? Type of checkers?)

8. Course Standardization (Number of students? Paid or unpaid? In-plant or outside?)

Total
man-hours______Materials $______Travel $_______

9. Other Production Work

Total
man-hours______Materials $______Travel $_______

GRAND TOTAL Man-hours______Materials $______Travel $_______

REMARKS: (Attach extra sheets if required) program can be scheduled into production with existing staff. In any case the statement of objective should communicate effectively and precisely the purpose of the project plan to all individuals who will assist in its preparation or interpretation.

Description of Work.

This is, of course, the bulk of the project plan and schedule. One of the advantages of the suggested project plan form is that this work-requirement portion has been broken into the various types of labor requirements that make up a typical project in programmed instruction. This method of breaking down the man-hour requirements greatly simplifies the costing process. Activities requiring similar levels of skill, and in general, similar rates of pay, have been grouped together. For the purposes of accounting and historical information, it is desirable to treat outside labor as man-hours rather than material whenever possible.

1. Course Planning. The number of hours required for course planning depends not only upon the length of the program but also upon the nature and complexity of the material. A project that is primarily research oriented will ordinarily require more planning time than a straightforward training program. A subject that has never been taught before or that is being developed in an area where the present training materials or methods are very poorly organized will, of course, require more time than a program based on an effective training outline. More time should be allowed for planning an abstract, highly verbal, or symbolic program than for a program covering mechanical, mathematical or procedural skills.

Included in the estimate of hours required for course planning should be the time required for the preparation of the Project Plan and Schedule, and for travel and necessary research. It is preferable to calculate consultant or outside man-hours separately from in-plant man-hours.

2A. Writing and Editing— Text. The length and complexity of the material also affect the amount of time required for writing and editing. In addition, the individual characteristics of the writer seem to be far more important in estimating time required than the individual characteristics of workers in other areas of programmed production. Polished professional program writers have shown writing speed variations from less than one-half page per hour to one and a half pages per hour. These rates are averages for complete projects and not day-by-day variations. On a day-to-day basis, the variation is much greater. Some days may result in a very large output; however, such days are usually preceded by several days of planning or consolidation in which there is little or no measurable progress.

When writing is separated from editing, the project averages indicate that there is wide individual variation among writers; some writers tend to produce consistently faster than others. However, when editing time is included in the project averages, these differences are materially reduced. This could be interpreted to mean that the slow writers spend more time in polishing their material before setting it on paper, thereby reducing the editing and rewrite time required. It could also be interpreted as a reflection of the effect of the normal distribution curve; that is, as more individuals' variations are averaged together, the result becomes more and more average.

Regardless of the factors involved, historical experience indicates that when the writing and editing time of a large number of writers and editors on a large number of projects of varying complexity and length is averaged together, the result turns out to be about two hours per page for writing and editing. This average can be used as a starting figure, which the project planner will modify according to the individual characteristics of the program under consideration.

The second important element in estimating the man-hours required for writing and editing concerns the number of pages or frames of material. Most intrinsically programmed materials will undergo expansion on the order of ten-to-one when put into program format. That is, one page of text will result in ten or more intrinsic pages or frames. For extremely technical or abstract subjects, the ratio may be closer to twenty frames per page. If this approach to estimating man-hours is used, the procedure is to multiply the number of pages in the present format by ten or more, depending upon the complexity of the material, and then multiply this product by the per-page rate. The final product is entered as the total man-hour requirement for writing and editing. If the program is to be based upon material that is well organized this task is relatively easy.

If the program is in a totally new area where previous material has not been developed, the alternative method based upon hours of instruction will have to be employed. As determined from experience in actual learning situations in which the total number of text frames was divided by the average study time, an intrinsic program of the proper level of difficulty and density for its purpose will provide approximately one hour of instruction per fifty pages of program. Thus, if the length of the course in hours has been determined, this number can be multiplied by fifty, and the product can then be multiplied by the established writing rate. This method can also be used as a rough check against estimates made by the previous method.

A third and more accurate method of estimating total man-hours required for writing and editing can be used in situations where the objectives and outline for the program have been established. This method requires that the outline be detailed to include the teaching points. Each point becomes a right-answer frame in the program. Most intrinsic programs average two wrong-answer frames for each right-answer frame. Thus, each teaching point represents approximately three intrinsic frames or pages of material. In this case, the number of teaching points in the outline would be multiplied by three and then by the average per-page rate for writing and editing.

"Materials" for writing and editing should include reference works, books, and other resources as well as stationery and miscellaneous supplies.

Travel requirements for the writing and editing staff are usually determined by the individual characteristics of the program and the skills and requirements of the writers and editors. Generally, a project should allow for two visits to the site by the writer.

2B. Writing Editing—Test Sequences. Test sequences for programmed instruction take on a variety of forms depending upon their purpose. For example, a cheat-proof test for the AutoTutor teaching machine will have only one question per page as compared to perhaps ten multiple-choice questions per page on a simple lesson test for a TutorText program. Yet, the writing and editing time for both tests will require about three hours per page.

Although a cheat-proof TutorFilm test page contains only one question, the question must be comprehensive and diagnostic. The whole test may contain only five or six questions, but these few questions must adequately test the student's knowledge of the entire lesson. Such questions are extremely difficult to construct. In estimating the frame count for cheat-proof TutorFilm tests, 20 images should be allowed for each question. The reasons for this will become apparent when the subject of test scrambling is dealt with in the section on Scrambling For Presentation on The AutoTutor Mark II:

Lesson tests for TutorTexts are very often application problems covering points in the previous lesson. These are relatively easy to construct, but since more of them are used, average writing and editing time remains about the same, three hours per page.

Criterion tests and pre- and post-program tests are probably the most critical tests the programmer is required to prepare. The writing and editing time would probably be higher for this type of test if the test construction time could be completely isolated from other phases of the program preparation, but since criterion tests cannot be separated from program planning and writing, the three-hour-per-page average is used for the preparation of criterion tests as well as for the other types of tests.

3. Other Editorial Work/Technical Review. This category of the Project Plan and Schedule is used to accumulate time estimates for miscellaneous and special editorial work. This would include consulting time for technical review of the material by either an outside expert or an expert within the organization ordinarily assigned to other duties.

This category is also used to accumulate time estimates for correspondence required in connection with the project and for the editorial preparation of instructors' guides, title pages, tables of contents and all editorial work related to the project which does not fall logically within one of the other categories.

4. Scrambling and Scramble Checking. Scrambling shows greater variability between types of materials than between individual scramblers. This variability makes scrambling time difficult to estimate if the format of the program has not been definitely determined. Fortunately for the project planner, scrambling and scramble checking account for a relatively small portion of the total project time and they are seldom performed by premium salaried personnel. Therefore, an error in estimating scrambling time does not carry the impact on the total project that would result from serious errors in editorial or writing time estimates.

There are several factors that influence scrambling estimates. When a book is scrambled, a short unit or chapter generally takes more time to scramble than a longer single unit. A fifty-page straightforward TutorText would require approximately one hour for scrambling. On the other hand, a similar TutorText of three to five hundred pages could be scrambled at the rate of approximately sixty pages per hour. The split-page format is economical and desirable in situations where the answer pages are generally short, but it requires more scrambling time. The rate of scrambling a split-page TutorText of approximately three to four hundred pages would average about ten to fifteen pages per hour. A short split-page program would require even more time per page for scrambling.

The time required to scramble and check TutorFilms also varies widely. Straightforward material with typical cheat-proof tests could be scrambled, paginated, and checked at the rate of approximately twenty to twenty-five pages per hour. The rate of scrambling TutorFilm involving numerous sub-sequences, washbacks (mandatory returns to earlier material), or express tracks (large jumps over material which the student has shown he already understands) will drop to approximately thirteen pages per hour.

Thus, it is important for the project planner to have some idea of the complexity of the material and the method of presentation. The use of this information in conjunction with the above guides should result in reliable estimates of the man-hours required to accomplish the chosen scramble. When man-hour requirements for this category are being computed, all frames—including title pages, cheat-proof test frames and table of contents pages—should be counted.

5A. Typing—Other than Mathematical. Typing can, of course, range from extremely simple to very complex. The average from program to program will also vary widely depending upon the nature of the material. The non-mathematical production typing of the Keesler Air Force Base Study averaged two and one half images per hour, and this figure might be used as a guide. The total number of pages estimated for the project, including title pages, index, etc., would be divided by the average of images per hour to find the man-hour estimate for this kind of typing.

5B. Mathematical Typing. Mathematical typing should be estimated separately from the text and other typing because it is a slower and more difficult process. Spacing, alignment, and use of special keys reduce typing speed. The potential for error and the need for accuracy are greatly increased in mathematical typing.

For estimating purposes, mathematical typing is considered to be a frame in which more than one half of the page involves the use of figures, formulas, or mathematical symbols. If averages for this type of production typing are not avilable within the local organization, then one and one half pages per hour may be used as an estimating standard. This figure was the production rate established on the preparation of the electronics TutorFilm prepared for the U. S. Air Force under this contract.

5C. Manuscript Reproduction. Multilith or mimeograph reproductions of the program are frequently used for preliminary testing and field testing. Determination of the format for testing will be discussed briefly in relation to the developmental cycle and covered fully in the section entitled "Program Validation." For the purposes of preparing the Project Plan and Schedule, it is necessary to know only the total number of pages in the program and the total number of copies that will be required.

The rate of reproduction for each page will decrease with increase in the number of copies required, but will increase with the length of the program. A working average for computing the time required is five pages per hour with fifty copies per page.

6A. Illustrating—Simple Line Art. For the purpose of preparing the Project Plan and Schedule, three types of illustration are included separately. The first is simple line art. This consists of charts and graphs and similar materials. Figures of this type can usually be completed at the rate of about three per hour.

6B. Illustrating—Complex Line Art. Complex line art includes cartoons, illustrations and detailed graphs involving numerous visual factors. These can usually be completed at the rate of approximately one per hour.

6C. Illustrating—Other. This category is used for photographs, complex drawings, and other illustrations requiring greater skill and more time than the first two categories. Each instance should be estimated on the basis of its own particular characteristics. Photographs and complex art work should be held to a minimum in most programs because this type of material does not reproduce well on the high contrast film required to produce legible text. Color films tend to be more satisfactory in this respect than a black and white film.

6D. Splicing. Splicing is used to add art work to the text material and to minimize retyping of final draft by removing errors physically and replacing them with corrected material. A useful estimate of splicing time can be obtained by computing the number of illustrations plus 10% of the text and multiplying this total by the rate of six pages per hour.

7A. Filming. Filming includes a check of the scramble in addition to the actual preparation of the master negative and the number of required prints. Most organizations will require filming services. Therefore, typical rates of outside vendors are supplied here.

The scramble check should be computed at an average of 20 pages per hour. Preparation of the master negative is computed at the rate of 50 cents per image (page). Prints are computed at the rate of 16 cents per foot. (One foot yields 16 images, 100 feet yields 1,600 images.) For example: Assume the program contains 112 images. 112 images divided by 16 images per foot equals 7 feet. Seven feet of program plus six feet of leader equals 13 feet per reel. Thirteen feet per reel times 7 prints equals 91 feet of printing. Ninety-one feet times 16 cents equals $14.56. Thus, in this instance, the 7 prints cost $14.56 plus the cost of the original negative. These rates are, of course, subject to change without notice on the part of the supplier.

7B. Film or Galley Checking. The galley check is the last opportunity to make corrections in the program before its release in printed form. All major changes should be incorporated before the galley proofs are prepared. In most instances, the printer or publisher permits a certain stated number of corrections to be made without charge, but additional corrections beyond this number usually incur severe penalties.

The same comments can be made with respect to film checking. Serious errors in the film can be very costly because the entire manuscript must be rephotographed in some cases. It is the film checker's responsibility to find what one hopes is not there.

Film or galley checking usually takes about one hour for each ten pages of actual text. Actual text excludes filler frames used to indicate to the student that he has arrived at a given point by mistake, and frames used to sequence the individual or return him to a previous point in the program.

8. Course Standardization. This category can be used to group together man-hour and material requirements for both the small-scale preliminary evaluation, which may be done using only a few students with rough draft material, and the large-scale field testing of the more polished version of the completed program. Three students are generally considered to be sufficient for the very early rough draft evaluation. Allowance should be made for a minimum of 50 students in the field-testing phase of the standardization. Although statistically valid results can be obtained from as few as thirty students, allowance must be made for drop-outs and individual cases that for one reason or another become invalid or useless.

The number of students required for testing should be multiplied by the estimated hours of instruction that the program will provide. To this should be added the man-hours that will be required for preparation and administration of the standardization program, analysis of the data, and preparation of reports to those concerned with the revisions of the material. If there is a great deal of variation between the pay rates of individuals conducting different phases of the standardization process, it may be desirable to list the different type of man-hour requirements separately.

Under "Materials" in this category should be listed space rental, teaching machine and equipment rental, and other miscellaneous supplies such as stationery. The cost of reproducing the materials for validation does not properly belong in this category but rather in category 5C, Manuscript Reproduction.

9. Other Production Work. This category is reserved for accumulating the costs of typing or printing reports, instructors' guides, student workbooks and other materials associated with the program.

Grand Total. When the estimates for each of the categories of the Project Plan and Schedule are completed, a total can be made of the man-hour estimates, the material estimates, and the travel estimates. At this point, a quick rough cost estimate (excluding overhead and general administrative expenses) can be obtained by multiplying the man-hour total by four dollars and adding this dollar figure to the material and travel totals.

The four-dollar-per-hour average cost figure will of course vary in regional labor markets and on projects involving extensive use of high salaried individuals. An example might be a program written by a team of physicists.

It would seem that the cost of a project involving extensive research would also be affected by the proportionately larger percentage of time required of the investigator. However, experience indicates that such projects usually involve a proportionately larger percentage of student testees or subjects receiving less than the four dollar average wage. This tends to cancel out the higher rates received by the investigator.

This shortcut costing estimate should not be the basis for preparing a budget or even for making a decision on the feasibility of the project. One of the purposes of the Project Plan and Schedule is to provide a structure for accurately estimating the many variables that can influence the cost or time requirements for a project.

Estimating Costs

The cost of preparing programmed instruction shows extreme variability. Some of the factors causing this variability have been discussed in connection with specific items in the preparation of the Project Plan and Schedule. In addition, there are other factors which are the result of interacting influences in the total programming process. The most important factor is, of course, the skill and thoroughness with which the program is put together.

A skillful writer after examining a number of programs could take a subject, chop it into sentences or paragraphs and prepare a manuscript which has the format of a program and looks like one. The cost would be based essentially upon the Project Plan and Schedule item of writing (less editing) and production typing. Although the product cost might be very low, the result would not be a program nor would it produce the results that are expected of a program.

The point is that the developmental process of programming is the essential feature in its effectiveness. Products that omit any of the essential steps are not truly programmed and represent a false economy at any cost.

On an industry-wide basis, the cost of programming appears to range from a high of about $40 per frame for difficult programs to a low of $4 per frame for simple linear programs. It should be emphasized that these figures represent the cost of preparing the manuscript and do not necessarily include general administrative expenses, profit, nor the cost of printing, publication, or filming. These latter costs show much less variability and tend to be comparable to the publication of any other book or printed material of like size and format.

The first step in estimating the cost of preparing the program is to translate the man-hour requirements into dollars. If the program is to be done as an in-house project, then the pay rate of the individuals assigned to the project will be the basis for determining these linear costs. If such data are not available, the actual costs incurred by U. S. Industries, Inc., Educational Science Division in the preparation of the programmed material under this contract may be used as a guide. These are shown below.

Writing and Editorial	$4.81 per hour
Editorial Support	2.64 per hour
Typing and Production	2.70 per hour
Art	2.97 per hour
Filming	3.97 per hour

Writing and editing cost can show extremely wide variations depending upon the requirements of the program. The editorial support includes personnel performing the scrambling function, art-text coordination, copy editing, checking and pagination. Typing and production includes proof reading and associated personnel. Filming includes the labor of film checkers and technicians. The salaries of psychologists, consultants, subject matter experts, and personnel to plan, administer, supervise and analyze the field testing results are not included in the above figures.

When the man-hours have been converted into appropriate dollar values, there is still another important step in developing accurate cost estimates. This is a cost adjustment determined by the length of the program. Both short and long programs are more expensive to produce than programs of intermediate length.

The initial costs, which include researching, planning, outlining, etc., are proportionately higher for a short program than for one of greater length. A certain amount of preparation is required regardless of the length of the program and the longer program is able to spread these costs out over a greater number of frames.

On the other hand, extremely long programs are difficult and costly to revise. Programs, particularly intrinsic programs, are like jigsaw puzzles. Each frame interlocks with its adjacent frame. Each frame builds on concepts developed in preceding frames. To remove, revise, or alter one frame requires the alteration of a number of associated frames so that the pattern will not have a hole in it. As each additional frame is revised, further revisions are required on down the line. Thus, a revision or alteration needed in the early part of a long program can call for extensive and costly revisions in all the units that follow.

One way to minimize this chaining effect is to write lessons or chapters as independent units. This is not always a completely satisfactory solution, and it does increase the cost and length of the individual units since additional frames are usually required to cover the definitions, terms, and other material necessary to make the unit stand alone.

The man-hour cost constants shown in table 1 are offered as a guide for use with estimates of long and short programs.

TABLE 1

Man-Hour Cost Constants

Frames	Cost Factor
0 - 150	2
150 - 250	1.5
250 - 500	1.3
500 - 1,000	1.2
1,000 - 3,000	1
3,000 - 4,000	1.3
4,000 - 5,000	1.5

Multiply the total cost estimate by the factor indicated in the table above if the program is shorter than 1,000 frames or longer than 3,000 frames. For example, using the table above, the cost estimate on a 600-frame program would be increased by 20%.

The final step in estimating the cost is to add the appropriate general administrative expenses. In most organizations, these are calculated as a percentage of the determined cost for the project.

In the final analysis, cost figures are no better than the data on which they are based. If a qualified person is not available for preparing the estimates on man-hour requirements, a qualified consultant should be secured. A $150 fee spent at this point can save many thousands of dollars that might be lost through an unwise decision based on an inaccurate cost estimate.

Completion Schedule and Coordination

A completion schedule is another type of general summary used for discussion in decision-making only. From it, the progress record chart and other more definitive and precise schedules will be developed.

There are two general approaches to estimating the completion schedule. We will deal with the straightforward approach first.

A logical approach is to establish a starting date for the project, and add to this the man-hours required for course planning and writing and editing of the initial chapter or lesson. The translation of this man-hour total into working days will give the date of completion for the initial chapter or lesson. The translation into working days of the man-hours for planning and writing and editing of all chapters will give the estimated date of completion of the final chapter or lesson.

When the completion schedule for the technical evaluation is being set up, the total time allotted to technical evaluation should be distributed throughout the program. This distribution may not be a matter of simple division. The evaluation of the first lesson or chapter usually requires half again as much time as the average for all of the units or lessons. The bulk of the evaluation time generally occurs during the first half of the program; as the program advances, the technical knowledge of the writers increases and their communication with the individual doing the technical evaluation improves. This will, of course, vary with individuals and with different subject matter.

If more accurate criteria are not available, it is suggested that the date for completion of technical evaluation of the initial chapter be established by adding one and a half times the average hour allotment for technical evaluation to the completion date for writing and editing of the first chapter. If the person doing the technical evaluation is a consultant located apart from the writing staff, transit time should be allowed both ways. This data translated into working days would determine the completion date for the technical evaluation.

Establishing completion schedules for production and filming is done in a straightforward manner. However, some lost-time allowance should be provided each time a unit or lesson transfers from one department to another. This is particularly true if the various units are geographically separated or if editorial coordination is informal.

A second approach for establishing a completion schedule is to work backward from an established completion date. This is more apt to be the actual situation in most cases.

The first step in this procedure is to translate the total planning and writing time into working days to determine whether the established completion date is realistic. If the required completion date does not permit normal planning, writing and editing time, then consideration should be given to the feasibility of assigning different writers to individual lessons or chapters.

If it is feasible to have the individual lessons of the program written simultaneously by different writers, then the man-hour requirements for the longest single lesson or unit should be added to the man-hours for planning and the total will determine the completion date for writing and editing the chapter or lesson which will be finished last. The same process when applied to the shortest lesson will determine the completion date for the chapter or lesson to be finished first. It should be noted that in this instance, the first chapter completed may not be the first chapter in the program sequence.

Under these circumstances, it may be difficult or impossible for the technical expert to evaluate the material in the order it is prepared. In such cases, it may be expedient to move the first chapter or lesson written directly into typing and art production. The material should never go beyond the production typing phase before it is approved from the technical standpoint. The retyping of finished manuscript copy is relatively simple when compared to the problem of reworking material that has been put into scrambled format.

This system of simultaneous scheduling, resequencing of work flow, and dovetailing of work, is used to bring the project's total man-hour requirement into line with the required completion date. Many of the principles and techniques used in PERT analysis are applicable to the development of a realistic completion schedule and to its implementation.

The proposed Project Plan and Schedule should be reviewed by an authorized and qualified representative of the groups that will be responsible for the editorial, production, and accounting functions of the project. If other departments or agencies will be directly involved in the project, they too should have an opportunity to review the proposed Project Plan and Schedule. They should understand that their signature on the Project Plan and Schedule indicates not only their review but also their accord with the estimates related to their respective functions. In other words, they should be willing to accept responsibility for implementing the Project Plan and Schedule.

REPORTING PROGRESS AND RECORDS

The purpose of the present report is to detail procedures for the preparation of intrinsically programmed material. Therefore, only those forms essential to the actual preparation of programs will be discussed. Requisition forms, accounting forms, personnel forms, and forms required by specific agencies or departments will not be considered.

When properly used, some forms are extremely important to the efficient on-schedule preparation of program materials. Improperly used or overly complicated forms and records can actually impede work on a programming project. The number of forms used should, therefore, be limited to those essential for accumulating information that is actually needed and will be used. The value of the data that will be supplied by a form or record should be weighed against the effort and time required to prepare and complete the record.

Project Progress Record

The Project Progress Record is one form that should be considered essential to a project of any size. It is a simple method of comparing project scheduling with actual accomplishment. It provides the Project Director with a sensitive instrument for anticipating delays and bottlenecks and taking corrective action. It also provides the essential historical data necessary to evaluate the efficiency of functional units of the programming team.

A sample Project Progress Record is shown as figure 2. This form can, of course, be modified to serve the specific needs of a project or an organization.

Identifying information on the sample form is the project title, the project number, and the unit. Each lesson, chapter, block, or unit is charted on a separate Project Progress Record form. Instructor's guides and other associated materials are also charted on an individual Project Progress Record form. Thus, a ten-lesson TutorFilm program with a printed instructor's guide and criterion pre- and post-tests would require thirteen individual progress record sheets.

The target dates are established by using the Project Plan and Schedule and working backward from the completion date. The use of different kinds of estimated target dates (such as optimistic, anticipated, and critical dates) has not been found desirable because of their actual closeness in time. In the Project Progress Record form shown, Item 23, "Released," refers to the date the program is released for full-scale field validation. If the project includes additional work beyond this point, these other items would be added to the progress record form. Not every item on the sample form will be used in every project. For instance, a program in book format would move from Item 18, "Fifth Edit Complete," to Item 21, "Galley Proofs Returned," with appropriate allowance for work by the publisher. "Filming Complete" and "Film Check Complete" would be omitted. Conversely, a TutorFilm program would substitute Item 20, "Film Check Complete," for Item 21, "Galley Proofs Returned."

The man-hour data in the Project Plan and Schedule covers each category for the total project. In the preparation of the Project Progress Record, a proportionate amount of this time must be allotted to each of the units. Some knowledge of the proposed content of each unit is necessary to make intelligent judgments concerning this apportionment of time.

In most projects, there will be some lessons that require more art work than others. The first lessons of a project usually require more writing time than do the later lessons. Lessons involving a great deal of mathematical typing will move through production at a slower pace than lessons which are primarily expository text. All of these factors must be considered in establishing the target dates.

Another factor to be considered in establishing the target dates concerns the number of writers that will be assigned to the project. If only one writer is assigned, the lesson preparation must be scheduled sequentially. If a number of writers are assigned to the project, the continuity will be likely to be better if a given writer can be assigned a workload consisting of sequential units or lessons. This may not be possible if the writers were selected because they possess special and varied skills required by the nature of the program, and the units in which these skills are required do not fall in sequential order in the program outline. All these factors will influence the scheduling of the target dates.

During the preparation of the program, the Project Progress Record serves as a means of control over all of the related activities. For this reason, the master copy of the Project Progress Record should be kept up-to-date by the Project Director. In a small operation involving ten individuals or less, it has been found feasible to maintain the master Project Progress Record in the Project Director's office and to have a copy of the record for each unit to travel with the manuscript throughout the entire process. As each project phase is completed, the completion date and initials of the worker are inserted in the manuscript copy and it is returned to the Project Director's desk. The Project Director brings his records up-to-date, reviews the material if necessary, and passes it on to the next phase.

In very small operations, the Editor and Project Director may be the same person. If the time factors are not extremely critical, it may be feasible for the Editor to up-date his Project Progress Record at the time each of the five edits is completed. Although this does not provide a tight control over the project, the same degree of control is usually achieved through informal knowledge of the work of the participating team members.

In large operations where several projects are in simultaneous production, communication and control are more difficult. It is frequently not feasible to have each unit of the project pass across the Project Director's desk after completion of each phase. In such cases, a simple slip type of phase completion form can be supplied to each of the team members. When a phase is completed, the unit progress record is signed off and a phase completion slip is forwarded to the

PROJECT PROGRESS RECORD

Project Title ____________________

Project No. ____________________ Unit ____________________

	Target Date	Date Completed	Project Phase
1.			Project plan prepared
2.			Book outline
3.			Chapter outline
4.			Chapter written
5.			First edit complete
6.			Rewrite complete
7.			Technical review complete
8.			Second edit complete
9.			Rewrite complete
10.			Scramble complete
11.			Student review
12.			Third edit complete
13.			Rewrite complete
14.			Fourth edit complete
15.			Final typing complete
16.			Proof reading complete
17.			Art work complete
18.			Fifth edit complete
19.			Filming complete
20.			Film check complete
21.			Galley proofs returned
22.			Final corrections
23.			Released

Figure 2

Project Progress Record

Project Director. The Project Director can then review his project record log and update it as often as necessary to maintain control of the project.

In multi-project situations, it is also important that the individual project directors keep the supervisory personnel of each Department informed of any significant changes in the project schedule. This is particularly important to the Production Department where the same personnel are used simultaneously for typing, filming, etc., on several different projects.

The production workload can frequently be evened out by using items such as the instructors' guide, pre- and post-tests, and title pages to fill in production gaps caused by unexpected delays in the program preparation.

Writer's Progress Record

This form is usually not essential in small operations, but it is valuable in multi-project operations, in situations where writers do not work in-plant, or where liaison between writers, editors, and project directors is difficult. It is also valuable where individual units are very long or require extensive writing time.

The form consists of a simple slip that identifies the project, the project number, unit or lesson, writer, the last pages completed at the time of the previous report, and the last page included in the present report. If this form is used, it is generally sufficient to use it for reporting once a week. Each time the writer's progress report is completed, the writer should record the page reference of the report submitted as the last page on the blank form to be used the following week.

Besides providing the Project Director with the information he requires, the Writer's Progress Record facilitates the accumulation of writing averages for individual writers on various types of projects. This information is frequently valuable in staff evaluation. The system may also be used to induce a subtle pressure on the writer to maintain a satisfactory level of production. This is particularly important in situations where the writers work at home or away from the plant site. It tends to even out the writer's level of productivity and to cut down on last-minute attempts to fulfill a writing assignment.

Program Record Form

The third essential form peculiar to program preparation is used for the accumulation of data during the testing phases of the program. The use of this form, an example of which is given as figure 3, will be discussed in greater detail later in this report. In each case, the form should be adapted to the specific program use intended.

There are certain essential items of information that are necessary regardless of the format used. Provision should be made for the date and the identification. Identification may be identification of the student, or of the program unit, or both.

The second essential of the form is the provision for recording the student's progress through the program and an opportunity for the student to indicate his reactions to and comments on individual pages or frames. Specific testing purposes may require additional information.

SELECTING PERSONNEL FOR PROGRAMMING

In the very early days of programming the opinion prevailed that programmers are born and not made. This was largely a reflection on the state of the art at that time. It revealed the fact that the skills required of a programmer had not been isolated.

It was, and still is, true that some individuals develop competence in programming much more rapidly than others. Some seem never to be able to develop the required skills. Part of the early problem resulted from attempting to isolate vocational backgrounds that would be reliable sources for recruitment. Experience has demonstrated that the vocational field is far less important in determining the success of a potential programmer than the personal experiences of the individual. Many of the intuitive processes previously used in the selection of programming personnel have given way to more perceptive techniques.

Program Record Form

Identification______________________________ Date____________ Page__________

Please record the order of your progress thru this program by writing the page numbers on the blanks below as you select them. If you are returned to a previous page record its number in order as often as you return. Add any comments you wish to make to the right of the page number.

Page	Comments

Figure 3
Program Record Form

The two major systems of programming, intrinsic and linear, originally approached the problem of recruitment in two very different ways. Both of these proved ineffective and costly.

In 1956, a group of psychologists met at Harvard University for the avowed purpose of "programming American education." As they were psychologists, they naturally assumed that a background in psychology, and particularly learning theory and more precisely operant conditioning, was essential to the task at hand. This viewpoint prevailed for a number of years and still has a dominant influence in some organizations.

But, the primary function of programmed instruction is communication. Thus, although psychologists concerned with programmed learning may be expected to contribute scientific advice based on their knowledge of human learning, this capability does not necessarily qualify them as effective communicators or program writers. Accordingly, the role of psychologists has gradually shifted to planning and advising, and other qualifications besides a knowledge of psychology are now considered more important in the individual who writes the program.

Intrinsic programming approached the problem of writer selection from an entirely different viewpoint. Intrinsic programming was invented by a psychologist, Norman Crowder, but his orientation was based upon communication theory and systems development. This viewpoint focused on the employment of techniques that would improve communication efficiency. Instead of seeking a particular training or background orientation, therefore, the Crowder group selected persons on the basis of their ability to communicate in writing. However, even this proved to be an inadequate criterion.

A Sample Program

Many individuals who had successfully demonstrated their ability to write effectively proved quite inadequate as programmers. Therefore, a policy and procedure for selection was evolved that, in the final analysis, is still unsurpassed. The procedure is simply to give the prospective programmer a good sample program to be used for study and as a model. The individual is then asked to prepare an original sample program on a subject of his own choosing. Surprisingly few aspirants are successful in doing so. It is this experience that led to the conclusion that programmers are born, not made.

This method of selecting personnel has proven to be somewhat inefficient, however, and if used exclusively, it would severely limit the use of programmed instruction by restricting the manpower available for the preparation of programs. The skills and techniques of programming are so diverse and subtle that very few individuals can be expected to acquire them all by the process of observation and imitation.

The need for an expanding programming staff led to experimentation in the field of teaching programming skills. An analysis of the programming task was performed and the results applied with considerable success by U. S. Industries, Inc., Educational Science Division, in two experimental schools in 1961. This and subsequent school experience has resulted in an accumulation of extremely useful information for selection of programming personnel. However, most of the progress has been in the identification of required skills and much less has been done about suitable instruments for selecting individuals possessing these skills. In the following discussion a considered approach based on this experience and present information is presented.

The first step in selecting personnel for programming is to determine what characteristics or qualities are required. Experience has shown that the mechanics of form and format can be taught to any person of normal intelligence. The psychological principles upon which programming is based can also be taught in a straightforward manner. The skill in the use of these principles increases with practice and is subject to wide individual variability. The mechanical skills of writing including grammar, punctuation, and spelling can be taught with moderate success, but except in rare instances should not be considered as a legitimate function of program training. Rather, these skills should be considered a prerequisite to selection.

It has also been demonstrated that the techniques of defining objectives in measurable terms can be successfully taught and the procedures used in performing task analysis can be successfully communicated. Skill in outlining can be developed. Students are usually able to learn ways of increasing their creative writing efficiency. The strategies of various programming techniques are amenable to the teaching process. Other programming skills that can be taught to any normally intelligent person include terminology and definitions, techniques of frame analysis, methods of sequencing materials, the procedures of editing, the principles of scrambling, methods of analyzing student error rates, procedures to be followed in revising materials, typing and illustrating specifications, photographic requirements, proof-reading techniques, control-coding, and the use of student progress charting.

With respect to administrative skills, it has been found that individuals can be successfully trained in how to plan a project, how to estimate costs, how to develop and use reports and records, how to construct and evaluate criterion tests, how to design evaluation studies and experimental research, how to plan and administer pre- and post-testing, how to analyze terminal behavior, how to successfully implement programmed instruction, and how to determine training costs.

Even with this knowledge, a person may fail to become a successful programmer because he lacks certain personal characteristics or aptitudes. We will now examine some of the qualities that the individual should possess before undertaking the training designed to give him specific techniques and polish.

Intelligence

The first of these prerequisites is intelligence. More specifically, it is verbal intelligence. A programmer must be a good learner before he can be a good teacher. If he is even moderately active in the programming field, he will be called upon to go outside of the area of his particular expertise in his programming efforts. In order to program unfamiliar material, he must be able to understand thoroughly and quickly. The cost of programming and the required completion dates usually result in time pressure on the writer. To be successful, he must be able to quickly generalize the concepts and principles from the material he has to study. He must be able to think at an abstract level but communicate at a concrete level.

If intelligence test data is available, an IQ range between 120 and 140 is probably a desirable criterion range. Below 120, the individual may tend to think in rather concrete terms. This may result in difficulty in assimilating the subject to be programmed. This is particularly true if the subject matter is not presented in well-organized form and is not thoroughly outlined. It may also result in programs that overemphasize the concrete aspects and obscure or avoid the principles or abstractions involved. At the upper extreme, the individual may program at an abstract level that fails to communicate effectively with the students for whom the program is intended. This tendency may be offset by the factor of empathy to be discussed later. However, it is the opinion of this writer that empathy is a relative characteristic determined to a large extent by similarities between the individuals in which empathy exists. In other words, an extremely intelligent individual may be capable of empathizing with a person of like intelligence but limited in his ability to empathize with individuals greatly different in respect to this capacity.

If intelligence test data is not available, the alternatives are to secure this data through testing, to make a subjective evaluation based on other criteria, or to ignore intelligence as a factor. Securing test data is not always feasible and when possible must usually be handled with tact. Even when handled tactfully, intelligence tests are generally time consuming and usually looked upon as a bothersome chore by the testees.

For the purposes of selecting personnel for programming, a rather unusual form of verbal intelligence test has been found helpful. This is the Full Range Picture Vocabulary Test by R. B. and H. S. Ammons, published by Psychological Test Specialists, Box 1441, Missoula, Montana. The test requires only five minutes to administer and, because of its novelty, it overcomes the reluctance expressed by most testees.

The test consists of a series of 16 plates or cards, each of which displays four line drawings. The testee is asked to point to the drawing on the plate which best represents a particular word. He is instructed not to guess, and he is asked to define words or point again later in the test on words the examiner thinks he has guessed. Words are given until three "point levels" are passed consecutively and three failed. The point levels are given on the answer form after each word and represent approximately the mental age at which 50% of a representative population would fail the word. The words move from concrete to abstract. The stimulus words with their point-level assignment are not visible to the testee at any time.

An experienced administrator can usually determine the approximate point level at which the testee can operate by about the third plate. Thus, the administration of the test can be expedited by omitting the words that the subject will obviously be able to handle correctly. The test administrator begins with a word that he feels is within the range of the subject's vocabulary. If the word is missed, he can drop down the list three words for the next stimulus. If the second word is correctly defined, he moves up the list until the subject has completed three successive passes and three successive failures. At that point, the test moves to the next plate.

Scoring is extremely simple and yields the mental age equivalent without reference to charts or scales. The number of items answered correctly is counted for each card and entered in the lower right-hand corner of the test form. All items below the three passed consecutively are assumed to have been passed. Passes for all cards are totaled and this total represents the equivalent mental age. The scoring process can usually be completed within a minute after the test has been administered.

If it is not feasible to test the prospective programmer, then mental ability may be inferred from other evidence. The best evidence other than testing is probably academic achievement. Although there are great differences in the mental requirements of schools and colleges throughout the country, it is usually safe to assume that an individual who has completed the B. A. or B. S. degree in the normal four years of time will probably possess an IQ of 110 or higher. It is generally agreed that an IQ of about 120 is required for successful completion of a graduate degree. Of course, many intelligent individuals have not had the opportunity of a college education. Less accurate methods of inferring intelligence are based on reading interests, vocational advancement, and verbal fluency. However, even a skilled interviewer is on dangerous ground if he makes a decision to eliminate a potential programmer on such tenuous evidence. Impressions of this kind should be used only to support a decision made on other criteria, which will be discussed below.

Verbal Facility

Programmed instruction should represent the highest level of perfection in verbal communication. Thus the ability to communicate through the written word is a prime requisite in selecting personnel for programming. We frequently meet individuals who can "talk a good line" but are ineffective in expressing themselves in writing. Therefore, the only feasible way to evaluate this requirement is from a sample of some original expository material written by the subject. A good procedure to use is to ask the subject to explain how to do something in 150 or 200 words. The topic should, of course, be familiar to the subject and may be simple or complex. An example might be how to lace and tie a shoe.

There are two elements of written communication that are important in programming: concept communication and readability. Concept communication determines to a large extent the program's effectiveness in achieving the desired terminal behavior and the objectives. Readability has a more subtle but important role in the program's effectiveness. A program that lacks readability will be frustrating or boring. Readability can be objectively measured but evaluation of concept communication is, at this stage of the art, still quite subjective.

Although the evaluation of a written sample of concept communication is essentially subjective, there are fortunately some aids to minimizing subjectivity. If the student could be asked to state the objective of his sample dissertation in measurable behavior terms, then the concept communication could be evaluated on the basis of the reader's ability to perform. This, of course, is not feasible because the potential programmer cannot be expected to perform a task which will become a part of his later training. This does give a clue to a good approach, however.

Evaluations are always relative to some sort of norm. If the student's writing assignment is on a subject about which there is general agreement as to the correct or incorrect performance, then a stable standard is available for the evaluation of concept communication. For example, it would not be difficult to get a group of individuals to agree on whether or not a shoe was correctly laced and tied. If describing this procedure in writing was required of the student, it would not be difficult to determine whether he had included all of the steps in the proper sequence and in adequate detail. The principle to be followed here is to assign a topic upon which there is agreement between the person doing the writing and the person doing the evaluating as to the facts. This is most likely to occur if the topic is thoroughly familiar to both parties.

The method of determining readability is much more standardized and can be applied to the same sample material. Generally, 100 words or more of material is necessary for establishing a valid index of readability, and for this reason the writing assignment should consist of 150 to

200 words. Since the procedures for determining readability have been thoroughly worked out and are in print, this report will not attempt to duplicate material but will merely list the sources available. Most systems are based on a count of total words and total syllables, which yields an index either by means of a mathematical formula or by reference to a chart. These sources are listed below:

1. The Lorge Formula for Estimating Difficulty of Reading, by Irving Lorge. Bureau of Publications, Teachers College, Columbia University.
2. A Table for Rapid Determination of Dale-Chall Readability Scores, by George R. Klare, University of Illinois, Urbana, Illinois.
3. A Formula for Predicting Readability, by Edgar Dale and Janne S. Chall, Bureau of Educational Research, Ohio State University.
4. Predicting Readability Levels, by Morton Botel. Follett Publishing Co., Chicago, Illinois.

Empathy

Empathy is the imaginative projection of one's own consciousness into another human being. In programming, it is the programmer's ability to view the material from the orientation of the student and to structure the material and test the questions in reference to this orientation. It is the quality that causes the student to experience the feeling that he is in conversation or two-way communication with the program writer. It is the quality that led Dr. Wayne Gustafson to the conclusion that successful programmers must have a certain inherent aptitude which cannot be produced by training.

When empathy was identified as an essential prerequisite in successful programmers, it was assumed that this quality would be found highly concentrated in the teaching field. This has not proven to be the case, although it is probably a common characteristic of our most successful teachers (when teachers are distinguished from lecturers). Current evidence seems to indicate that it is pretty generally distributed (in small quantities) among members of all occupations and endeavors, including business executives, counseling psychologists, social workers, first sergeants, and factory foremen. It is not necessarily a quality of leadership, although its presence usually makes the leader more effective. There is some evidence that empathy is more frequently found in individuals who are more than usually introspective.

Unfortunately, there is no precise method of selecting or predicting empathy in the potential programmer. A clinical psychologist or psychometrician can diagnose this factor with fair assurance, using projective techniques such as the TAT. This procedure is generally not feasible and, even if possible, is not recommended because empathy is a relative quality. It has been observed that in some instances programmers who demonstrated empathy in some subject areas or with certain learner groups tended to lack this quality when assigned to work on projects of a very different nature. One example, already mentioned, is that of the high IQ individual who might be able to empathize with other individuals of similar intellectual level but finds himself incapable of showing empathy for the slow learner. Observations of such "selective empathy" have led to the conclusion that empathy is not a quality that a person either possesses or lacks universally. Rather, it seems to be a quality that depends in part upon the individual's psychological makeup and in part upon his background experiences. It has been observed further that programmers who tend to possess a high capacity for empathy have had fairly wide and varied backgrounds. From this, one might tentatively conclude that, given an introspective (not introverted) personality, the individual will display more of the characteristics of empathy in relation to students coming from environments similar to his own than in relation to students from dissimilar backgrounds. To state it succinctly, a shop foreman writing in colloquial English could probably prepare a more effective program on turret lathe safety, other programming factors being equal, than either the engineer who designed the lathe or the English professor who writes in polished phrases.

Although it may be difficult or impossible to select personnel on the basis of some universal quality called empathy, it is feasible to select personnel with backgrounds characteristic of the intended student population. When this is not feasible, the individual should be given the opportunity to experience first-hand the pertinent problems and environments of the student population.

Question Construction

The most time-consuming element in writing intrinsic program frames is the construction of the question. The reasons for this will become obvious in the section dealing the multiple-choice questions under "Programming Technology." The point of concern here is the fact that some individuals experience a great deal more difficulty in preparing good questions than do others. The subject is a part of the regular curriculum of U.S. Industries, Inc., Program Writers' Schools and although all the students show improvement with training and practice, some are never capable of fully incorporating the elements of testing the communication, testing the central concept, testing the student's ability to apply knowledge, and diagnosing identifiable types of errors within the framework of a single question. Since these functions of the question are essential elements in an effective intrinsic program and since experience indicates that training is more successful in polishing the technique than in teaching it, it is of advantage in selecting personnel for programming to evaluate the individual's potential with respect to this capability.

Research on an evaluation instrument for this use is being done in conjunction with U. S. Industries, Inc., Program Writers' Schools. Although the data accumulated so far is not sufficient for statistical analysis, the instrument and suggestions for its use are included in this report.

The test is called the PDQ Inventory (preparing diagnostic questions) and is shown as figure 4. The inventory consists of a stimulus paragraph and instructions to perform three specific tasks. The first instruction is to state the major concepts in the paragraph. The second is to use the most important concept as the basis for construction of a multiple-choice question with at least three alternatives. The third task is to use the second most important concept as the basis for a multiple-choice question with four answer alternatives of one word each. The subject is required to indicate his starting time, completion time and total working time.

The PDQ Inventory is an attempt to identify two important characteristics of successful intrinsic programmers: the ability to abstract concepts from written material, and the ability to construct multiple-choice questions to test these concepts. The inventory is based on the following rationale:

1. The construction of multiple-choice questions is the most time-consuming (and therefore expensive) aspect of program writing.
2. Some people can frame multiple-choice questions more quickly and easily than others.
3. People with highly developed skill in constructing questions may not be good programmers, but programmers without this skill will be slow and unproductive.
4. A test can be constructed to identify objectively a well-developed skill for constructing multiple-choice questions.
5. The use of such a test as a screening device in association with other selection techniques will tend to reduce the risk of investing in unprofitable personnel.

The test is not intended or used as a substitute for a sample program. Instead it is to be used as a supplement to provide valuable information which can be used in counseling prospective writers. The test is scored on the following criteria:

1. The subject's ability to follow instructions.
2. The subject's ability to abstract ideas from material.
3. The subject's ability to construct multiple-choice questions.

The subject is scored A (excellent), B (above average), C (average), D (below average), or E (poor) on each of the three criteria stated in the previous paragraph. The scores are based on a comparison with standards developed for this inventory.

The standard scores are subject to continuous revision, but were initially developed in the following manner:

1. The test was given to a group of staff writers at U.S. Industries, Inc., Educational Science Division.

THE PDQ INVENTORY

Name____

Address______________________________________

________________________ Phone________________

Date____________________

This inventory should be completed in one uninterrupted session. When you are ready to begin, record your starting time in this space:

____________p. m. a. m.

Now read the paragraph below:

The attitudes of typical students are useful to programmers. John is a typical student. He does not like to make mistakes, but he usually profits from them if he is corrected immediately. He likes to learn by doing in situations where he can set his own pace.

1. On the back of this sheet, state the major concepts contained in the previous paragraph.

2. Using the concept you consider to be most important, write a multiple-choice question with at least three (3) alternatives that will test the student's understanding of that concept.

3. Using the concept that you consider to be next in importance, construct a multiple-choice question with four (4) answer alternatives of one (1) word each.

Record the time of completion in this space:____________p. m. a. m.
Subtract your starting time from your completion time and record it in this space:
________________working time.

You are to be congratulated upon completion of a task that is quite difficult for most people.

Figure 4

The PDQ Inventory

2. The results of this initial administration were arranged in rank order on each of the criteria.

3. On each of the three criteria (ability to follow instructions, to abstract concepts, and to construct multiple-choice questions), the best response and the poorest response were selected. Then, the response most nearly equidistant between the best and the poorest was selected. These samples represented the original judgment criteria. Copies of these records along with the scores they represent on each of the four criteria were then used as the basis for evaluating subsequent test results.

When a prospective writer completes the test, the results are compared against these standards. If the prospective writer's results are better than the standards being used as "excellent" on a given criteria, then the previous "excellent" standard is discarded and a copy of the new record is used as the new standard of excellence on that criterion.

If the prospective writer's results are poorer than the record currently used as the standard E (poor), then the new record replaces the old standard on this criterion.

If the prospective writer's results are poorer than the best but better than the poorest on a stated criterion, then they are compared with the sample considered average on that criterion. If they are better than the present average, the grade becomes B. If they are poorer than the average but better than the poorest, then the grade becomes D (below average).

This method of evaluation causes the deviation on each of the criteria to spread continuously, and by this method, the sample becomes more representative of the true population.

No direct use is made of the time data in scoring this inventory. The primary purpose in requesting time data is to place the student under pressure to work as rapidly as possible without interruption. The student, of course, does not realize the time is not a scored factor.

The time data is valuable in drawing subjective inferences regarding the prospective writer. It is important to note how the individual handles a time pressure situation. Some individuals respond well to pressure and this is desirable. Others tend to freeze up under pressure. Some tend to cheat on their time report.

The test paragraph was constructed to include the following basic concepts:

1. Errors are unpleasant.
2. We learn from errors.
3. Immediate knowledge of result is important in learning.
4. People learn by doing.
5. Self-pacing is important.
6. These generalizations are useful to programmers.

The above concepts are rationally considered to be basic. Irrelevant and secondary points include items such as "John is a student" or "The paragraph is about John." Most other points that have been mentioned to date are either fragments of the above concepts, combinations of the above concepts, or variations of the above concepts. Variations that do not lose the meaning are, of course, acceptable.

Recruiting

As stated at the beginning of this discussion on selecting personnel for programming, it appears that no single occupation or academic background produces the qualities necessary for successful program writing. U. S. Industries, Inc., Educational Science Division has recruited writers from such varied fields as the theater, the medical profession, social work, the publishing field, advertising, teaching, electronics, and homemaking. Some backgrounds do tend to produce skills useful in programming, but these must always be evaluated in relation to the composite requirements. For instance, one excellent programmer has considerable previous experience in preparing radio and TV commercial scripts, which taught her the skill of writing clearly and succinctly. However, this experience would not have been enough if the other qualities had not also been present.

Although experience has not indicated that there is any single reliable source of programming talent, it can be stated with conviction that a good programmer, regardless of background, must possess persistence, dedication, and a willingness to work hard.

PROGRAM DEVELOPMENT CYCLE

Now that pre-planning and the characteristics of successful programmers have been discussed, it is appropriate to consider in detail just what is involved in the programming process. In the development of programmed instructional material, certain activities must occur in a necessary sequence; hence there is an established Development Cycle (figure 5). The interrelationships of each of the phases of the Development Cycle will be treated in this section of the report.

Establishment of Requirement

The establishment of the instructional requirement has been previously discussed and is, of course, fundamental to any training program whether it be programmed or traditional instruction. In the case of programmed instruction, the completed program must be evaluated against this requirement to determine the program's degree of success. If the program fails to meet the desired criterion, either the program must be put through the development cycle again or the original requirement must be modified. Some of the factors that would affect such a decision include the relative success of the program, the cost of revision plotted against the potential increase in effectiveness, the number of individuals or man-hours affected by the training, and the time available.

Course Planning

Course planning, which is the second phase of the Development Cycle, has been discussed under the Logistics of Programming. It includes not only making the Project Plan and Schedule, but also establishing the objective, the course outline, the research design for testing and evaluating the program, and the instruments to be used in the evaluation. The course planning should clearly indicate the method and approach for meeting the requirements established at the outset.

In some cases, research and thought that go into the course planning will establish the fact that the original training requirement may not be a basic requirement. The planning effort may indicate, for example, that there is a more basic but previou unrecognized training requirement inherent in the situation. In such a case, the original requirement may, with the approval of all parties concerned, be revised to include the more basic need or may shift its emphasis to this previously unrecognized need. In either case, the original Project Plan and Schedule must be reviewed with the new requirement in mind.

Writing and Editing

When the course planning is completed and approved, the project is turned over to the Editorial Department where writing and editing are begun. It should be emphasized that this process is not comparable to moving a product from one phase of production to another. In all instances, the writing and editing staff figure prominently in the planning stages and have previously concurred on both the extent of their responsibilities and the time table upon which their activities are based. The Project Director works in close liaison with this group and every other group involved in each subsequent phase of the program development.

As each chapter is written, it receives its first edit. This is a programming edit that evaluates the material on the following criteria:

1. Whether the material follows the outline;
2. Whether the material is internally consistent and logical;
3. Whether the material follows the prescribed programming strategy; and
4. Whether terms to be defined in that unit have, in fact, been defined the first time they are used.

On the basis of this first edit, the material is returned to the writer for initial rewrite. At this point, fine points of style are not a major concern because the material will probably be

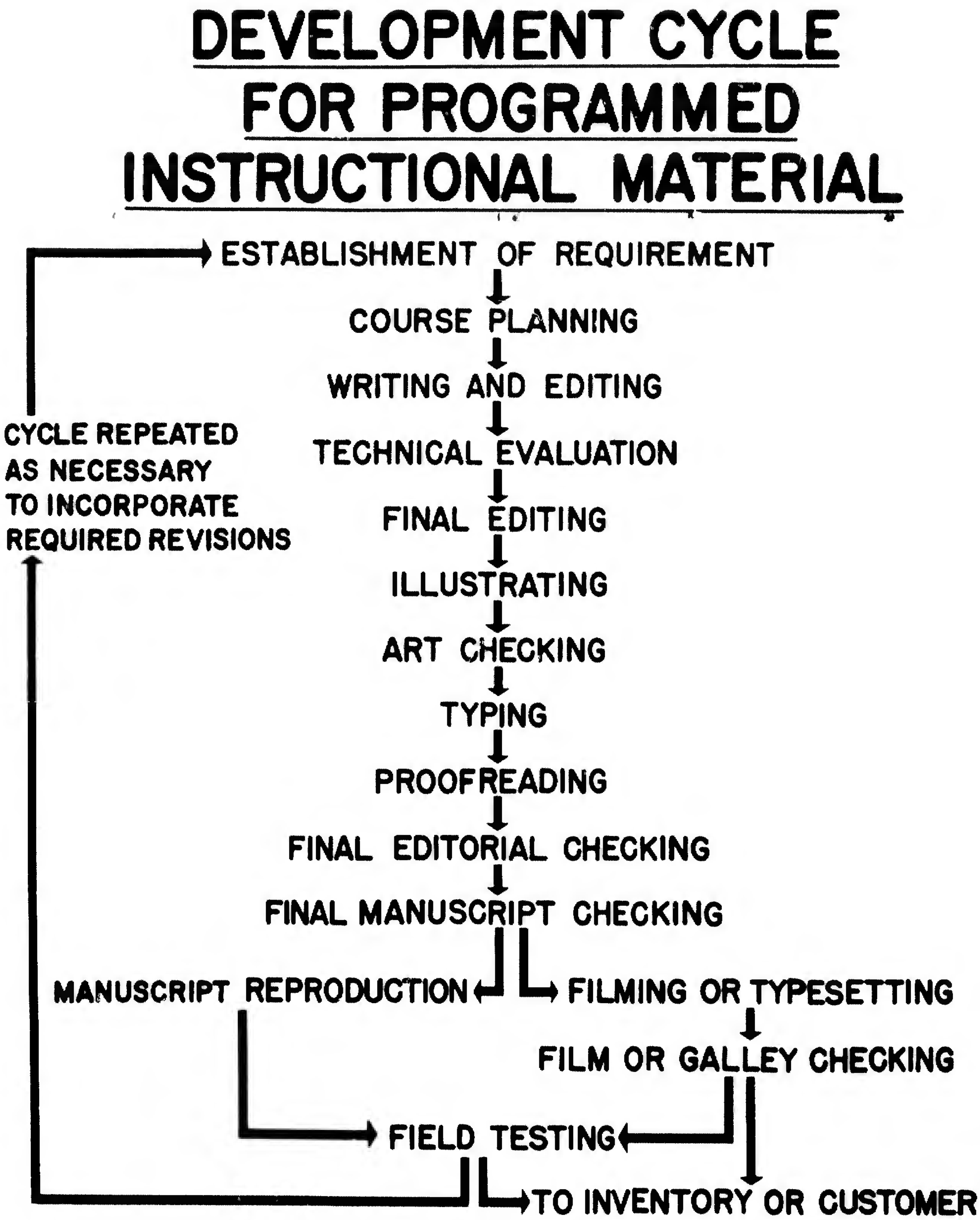


Figure 5

Development Cycle for Programmed Instructional Material extensively revised before the final draft.

Technical Evaluation

After the first edit and rewrite, the material is given to the subject matter expert for technical evaluation. This individual has previously assisted in the course planning phase of the project. His function at this point is to determine that the program is technically correct, that the answer alternatives are logical and consistent from a technical standpoint, and that the material provides the technical information necessary to attain the established objectives.

After the technical evaluation has been completed, the material is returned to the editor for his decision on the recommendations of the subject matter expert. Frequently, these recommendations go beyond simple evaluation of technical accuracy and include recommendations on programming strategy, reorganization of the material, or shifts in orientation. It is the Editor's responsibility to render decisions on which changes should be incorporated and to what extent the revisions should be made. This process is shown as Item 8 on the Project Progress Record (figure 2) and is the first phase of the final editing process shown on the Development Cycle.

Final Editing

The final editing phase of the Development Cycle includes Items 8 through 14 on the Project Progress Record: second edit and rewrite, scramble, student review, third edit and rewrite, and fourth edit.

The material is returned to the writer for incorporation of the recommendations of the subject matter expert which were approved by the Editor. After this rewrite is complete, the material is ready to be scrambled.

The type of scramble will depend upon the use of the program. If the final production is to be in book format, then only a book scramble will be developed. If the final format is to be on TutorFilm, then the material will usually be scrambled for both book format and TutorFilm format. The reason for scrambling TutorFilm programs in both formats is that the initial student review is usually based on a paper tryout. The exception to this is when the initial student review is to be conducted with a large group (30 or more).

After the student review is complete and the results have been analyzed, the recommendations are forwarded to the Editor. The Editor then determines the nature of rewrite necessary to carry out the recommendations resulting from the student review. The material is rewritten on the basis of these recommendations and returned to the Editor.

The fourth edit is the last step in the final-editing phase of the Development Cycle. This edit is used for the following purposes:

1. To determine that the last rewrite does accomplish the aims of the student review recommendations;
2. To determine that the programming strategy has not been materially modified in the rewriting process;
3. To perform a final copy and grammar edit.

Illustrating and Art Checking

The nature of the art requirements and the workload of the Illustrating Department will determine the exact point in the Development Cycle at which the Illustrating Department will receive the work order for the necessary art work. This point may be during one of the activities included in the final-editing phase but should be prior to the time at which the program goes to final typing. The dimensions of the final art work must be known to the Production Department to permit proper space allowances in the final production typing. As the art work is completed, it is checked by the person responsible for editorial coordination to ensure accuracy in every respect.

Typing and Proofreading

Each unit or lesson is proofread as soon as production typing is complete. After proofreading is finished, the art work is spliced into the camera-ready copy and the unit is ready for final editorial checking.

Final Editorial Checking

The final editorial check ensures that all art work agrees in every detail with the copy it illustrates, that all revisions have been incorporated in the proper manner, and that all pagination and other mechanical details are correct.

Final Manuscript Checking

After the final editorial check is complete, the Production Department performs a final manuscript or scramble check. From this point, the material will be considered ready for manuscript reproduction or filming or typesetting, depending upon the method of field testing that has been selected.

A TutorFilm can be field tested in manuscript format, although there are several reasons why the film format is to be preferred. The first of these is that TutorFilm is less expensive than the book format in field testing involving large numbers of students. The second and most important reason for preferring TutorFilm in field test trials is that the AutoTutor Mark II is capable of exercising a number of controls over the student which are not available with the scrambled book format. These controls ensure that the student follows the programmed sequence through the material and does in fact learn each concept before proceeding to the next. The third reason for choosing TutorFilm as a preferred method of field testing is that the recording systems fitted to the AutoTutor Mark II are less subject to error than progress records maintained by the student as he attempts to concentrate on the program material. The mechanical recorders of the AutoTutor Mark II also are less likely to slow down the student's progress through the program. Thus, more reliable time data can be obtained.

The results of the field testing are then subjected to statistical analysis, and this analysis becomes the basis for the decision to release the program or repeat the development cycle. With this brief overview as background, the technical aspects of preparing programmed instructional material will now be presented.

ESTABLISHING TRAINING OBJECTIVES, CONTENT, AND SEQUENCE

Objectives, like the weather, make an interesting topic of conversation but prove difficult to shape. Objectives are not new on the educational scene, yet it took the advent of programmed instruction to focus attention on their importance. Objectives are a way of communicating the intentions of one individual to others. In terms of programmed instruction, it is the programmer's means of communicating the changes he proposes to initiate in the student.

Presume for a moment, that it is the objective of this report to make you, the reader, a successful intrinsic programmer. Does this mean that the objective has been achieved when you prepare your first sample program or when your first program is published or when your first royalty check is received? Which criterion for success would satisfy you? Would a professional program editor or a writer of five published programs agree with your definition of success for achievement? The objective does not communicate clearly because success is a vague term meaning different things to different people. Suppose, however, that the objective had stated that you would become a successful intrinsic program writer with a royalty income of $10,000 per year within three years after studying this report. At the end of three years, there would be no question on the part of you, your editor, or any other person as to whether this objective had been reached.

Objectives communicate clearly only when they are stated in such a way that they mean the same thing to every individual for whom they are intended. The key to communicating objectives successfully is to state them in measurable terms. In other words, objectives should include not only the intentions of the programmer but also the unit of measure by which achievement is to be judged. As an adjective, objective means detached, impersonal and unprejudiced; objectivity is achieved by stating the unit of measure that will be used as the criterion. The task analysis is the most practical means of accomplishing this aim.

Analysis of Desired Behavior

Establishing the program objectives usually begins with rather general statements of what the training is to accomplish. In figure 6, Model for Program Planning, these are shown as the basic objectives near the apex of the triangle. These may not necessarily be directly or objectively measurable.

When the general definition or objectives for the training have been determined, then the programmer proceeds to define the tasks of the trainee more specifically. Generally, a subject matter expert should be used to assist in this phase of the task analysis.

The result of this process should identify the elements or individual skills required to master the task; identify the contribution of each element to the total operation of the task; and establish the frequency of occurrence of each element in the performance of the task. Efforts should be made to exclude any elements that do not contribute to the performance of the task.

Each of these individual skills will possess a theory element and a practice element which require identification. The theory and practice elements require separate approaches in the preparation of the programmed material. In figure 6, Model for Program Planning, these are shown as the secondary objectives. They represent skills or knowledge that are essential to the basic objectives and that can be measured.

Specification of Objectives

The programmer is now in a position to state objectives in terms of measurable behavior. When we speak of behavior, we imply activity. Activity means that the student is "doing something." Good secondary objectives state what the student will be doing and how his action will be identified and measured.

There is no way to measure or identify mental processes directly. Thus, there is no direct way to measure learning. However, learning that does not result in changed behavior is of little value and so the problem of measuring learning directly is rather academic.

Since effective learning does result in changed behavior, the problem becomes one of eliciting the behavior desired and then providing an objective means of measurement. The method used to elicit the behavior will depend upon the nature of the task. If it is feasible to require the student to perform the task and then measure his performance directly, this is preferable.

In some cases, the task may require too much time or equipment to be measured directly. Other conditions may also make it unfeasible to measure performance under actual conditions. For example, it is not feasible to require a student completing a program on mouth-to-mouth rescue breathing to demonstrate his proficiency by actually resuscitating an individual who has stopped breathing. In such a case, the performance to be measured must be based upon simulated conditions. Even here, the specific skills required in each phase can be demonstrated and objectively measured.

It was stated at the beginning of this discussion that the objectives should communicate the intent of the program. We emphasized the role of an objective unit of measure in assuring good communication. The semantics of the terminology is equally important. Some words are more ambiguous and open to misinterpretation than others. Words like know, understand, appreciate, believe, enjoy, and grasp frequently mean different things to different individuals. Such terms should be avoided in phrasing the objectives.

It is sometimes necessary to use such terms in the formulation of the basic objectives but they should be clarified and restated in the secondary measurable objectives. For example, a programming unit might be assigned the responsibility for preparing a program that will "develop an attitude of respect in enlisted personnel towards commissioned officers". This might well be the desired goal of the program but it would certainly not qualify as a measurable objective stated in behavioral terms. Through the process of task analysis, it would be necessary to specify behavior or information that contributes to or is indicative of "an attitude of respect." When all of the elements of this attitude have been defined, behavioral criteria can be identified. For instance, acts of courtesy would probably be one major area. These would be further broken down into

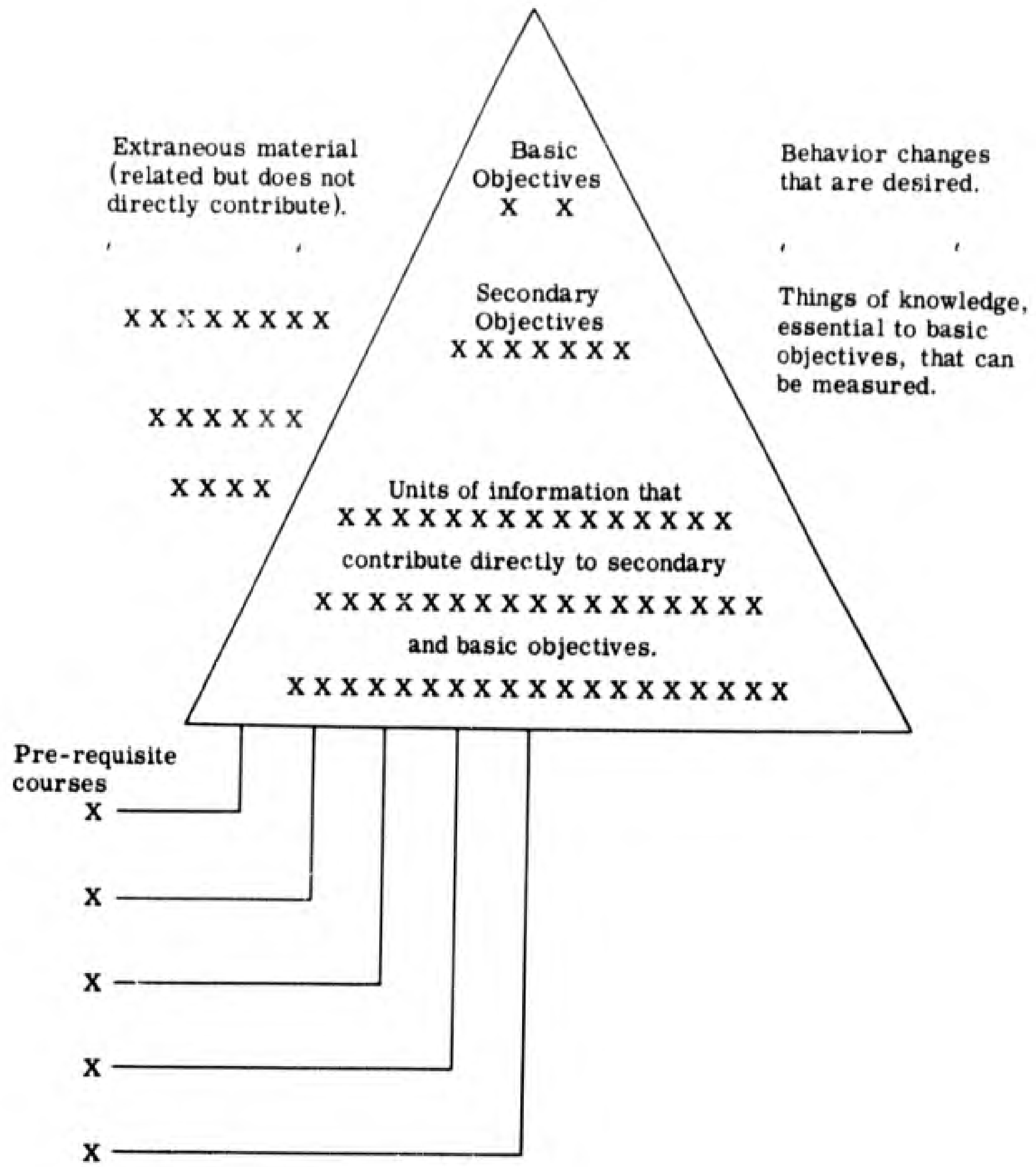


Figure 6

Model for Program Planning

specific acts in the presence of an officer, specific acts in the absence of an officer, etc. These specific acts would then suggest objective methods of measurements.

Avoidance of ambiguous terms like attitude and acceptance will materially simplify the task analysis and the preparation of measurable objectives. Additional guidance in establishing program objectives will be found in:

> Preparing Objectives for Programmed Instruction, by Robert F. Mager. San Francisco: Fearon Publishers, 1962.
>
> Programmed Instruction: A Manual of Programming Techniques, by Dale M. Brethower. Chicago: Educational Methods, Inc., 1963.

Evaluation of Objectively Relevant Program Materials

The basic and secondary objectives have now been determined and stated in terms that are agreeable to all parties concerned. The task analysis has been completed and the skill elements have been identified. The programmer is now ready to begin his detailed program outline. In doing this, he uses a criterion of objective relevance to determine what properly belongs within the framework of the stated objective.

In programming almost any topic, some aspects of the subject will seem to require inclusion because they are interesting, because they have traditionally been a part of the subject or because they seem to be logically connected with it. However, they may not pass the criterion of objective relevance. For instance, in the hypothetical program intended to teach the skills of rescue breathing, it might seem logical to include a brief history of artificial respiration. Yet, when the question is asked, "What does this contribute to the development of the skills required to perform a task?" the history of the subject no longer appears objectively relevant.

Each topic that is under consideration for inclusion in the program should be tested by the question, "What does this contribute to the established objective?" If a positive contribution to the objective is not apparent, then the programmer must either drop the topic from the outline or consider broadening the objective to include the topic. In some few cases, the process of outlining may uncover a topic that has been overlooked but is recognized as essential to the program. This is unlikely if the task analysis has been properly performed, yet the possibility must be recognized. Such topics must be weighed carefully because their inclusion will, of course, increase the length and cost of the program.

In such cases, the cost is not merely the cost of preparing the additional frames, but the cost in man-hours of all the students that will spend additional time studying these frames. Decisions to broaden or lengthen a program should be taken seriously and weighed carefully. If a proper task analysis has been performed, there should be very little justification for including material that is not objectively relevant.

Outlining the Program

It is difficult to separate outlining from establishing the program objectives. The interrelationship is close and important. In a sense, outlining is the process of further refining and reducing the secondary objectives. When this line of thought is carried to its extreme limit, the multiple-choice questions on each page become a measuring device for an item in the outline in much the same way that the criterion test becomes the measuring device for the secondary objectives.

With this relationship in mind, the objectives should be organized in a logical sequence before the secondary objectives are subdivided.

What is a logical sequence? It is the order that will produce the most efficient learning on the part of the student. Usually, this means moving from simple material to complex material, from concrete concepts to abstract concepts, or from what is known to the unknown. Surprisingly, this is not often the traditional sequence of the subject. Frequently, it is not the chronological sequence of the subject. Quite often, the logical sequence is to begin at the end of a task and teach backward toward the beginning. This is not always the logical sequence, but should always be considered as a possibility. This point is developed fully in the section on "Ways to Build the

Program" under "Programming Technology." The point here is that the programmer should not fall into the trap of assuming that any established sequence is necessarily logical or best.

When the sequential order of the objectives has been established, the next step is to determine what information should be given the student to accomplish each of the objectives. The task analysis is of material help at this point. Whereas the task analysis states what the student must be able to do, the outline becomes a statement of what the programmer must say to the student in order to cause learning to take place. At this point, it is better to have too many teaching points in the outline than too few, in order that no important elements may be overlooked.

When the teaching points for each of the secondary objectives have been established, this information should be organized in topical sequential form. In other words, all the teaching points on a given topic should be placed together.

Next, these teaching points should be reviewed to determine what information is required of the student to complete the program but not necessary to fulfill the program objectives. Some of the teaching points in this category may cover information that was included in the program assumptions concerning the student's background. If so, these teaching points should be discarded. If, on the other hand, the student is not assumed to know the information, yet it does not fulfill one of the program objectives, then it should be considered as appropriate material for a subsequence.

When the outlining is complete, each point in the outline should represent one right-answer page in the intrinsic program. Since there are generally two wrong-answer pages associated with each right-answer page, it should be possible to determine accurately the total length of the program from the finished outline.

Defining Terms and Concepts

Determining the terms and concepts to be defined in the program is best accomplished as a part of the outlining process. The procedure advanced here has been found to be effective; however, a less formal procedure may be used in situations where the programmer is thoroughly familiar with both his subject matter and the backgrounds of the students that will be using the program.

The formal procedure which minimizes subjective influences that might lead to omissions or errors in properly defining concepts consists of establishing two lists of terms. The first list contains terms that the programmer is certain will require definition within the framework of the program. These are technical terms or terms that are to be given special meaning with reference to the programmed subject matter. The second list consists of terms which are assumed to be understood by the student and doubtful terms which may or may not require definition in the program. Generally, these terms suggest themselves as the outlining proceeds. However, the list should be kept open and new terms added at any time in the programming process that the necessity for a definition becomes evident.

The next step is to construct definitions for the terms in both lists. The list of terms to be defined is set aside until the programming is undertaken. The list of assumed and doubtful terms is tested on a small group of typical students. Any terms in the second list that prove to be ambiguous should be taken from the assumed list and put into the list of terms to be defined in the program.

The remaining terms on the assumed list serve a second important function of communication between the various members of the programming team. The Editor, for example, may frequently raise a question concerning the need for defining a specified term in the program. If the programmer can demonstrate that it is safe to assume the student's knowledge of this term, then the matter is quickly settled.

The programmer should constantly consult the list of terms to be defined. As a rule, such terms should be defined the first time they are used in the program. As the programmer uses and defines each term, he should check it with the list. This will avoid the possibility of defining the term twice and will allow the programmer to use the term freely and with confidence after it has been defined. In many programs, it is also desirable to provide this list at the end of the program as a glossary.

There are many skills and procedures in preparing programmed materials that have been identified and can be successfully taught. Although, as was pointed out in the section of this report dealing with the selection of personnel for programming, an individual can develop skill in the technical aspects without ever preparing a successful program, the fact remains that this technology is important. For the purposes of this report, the subject will be discussed under a number of sub-topics. In practice, each of these is interrelated and effects the other functions. The skill in blending these techniques keeps programming within the realm of a practicing art.

Terminology and Definition

The field of programmed instruction has, like many vocations, developed its own professional jargon. Unfortunately, the terminology developed more rapidly than the technology, with the result that most of the jargon fails to communicate in a precise or scientific sense. Most of the terms probably find their widest use at conventions where they become a symbol of membership in a new and growing fraternity. However, to fulfill the requirements of completeness in a technical report on program preparation a thoroughly adequate list of terms and their definitions developed under the direction of Dr. Robert Orlando of the University of Minnesota is included. (Appendix C)

Psychological Principles of Intrinsic Programming

The rationale of intrinsic programming postulates that the basic learning takes place during the student's exposure to the new material on each page. The multiple-choice question is asked to find out whether the student has learned. The direct purpose served by the question in intrinsically programmed materials is to determine whether the student has understood the material he has just read. The reason for wanting to make this determination is that the process of symbolic communication is liable to error, and if there has been an error, or failure of communication, it is desirable to detect and correct the error before proceeding. Letting the student choose the next material he will see (by his act of selecting an answer to a multiple-choice question), makes it possible to detect and automatically correct any errors that occur.

Individual differences are an important source of communication breakdown. It is unrealistic to believe that in any practical situation all students come to the beginning of a program with the same amount of information. To accommodate these individual differences, diagnostic questions and remedial material are provided as an integral part of the program for those who demonstrate a need for such material; thus the intrinsically programmed portions of the course adapt to the individual differences among the students and allow the programmer to achieve the desired educational objectives with students of heterogeneous backgrounds.

Field theory is applied through the effective organization of the material, the step size, the level of difficulty of the questions, and the information provided on the wrong-answer pages. The individual's psychological field is increased only by relating new experiences and new information through experiences, concepts, ideas and information already within the field. In intrinsic programming, this is done by moving from the familiar to the unfamiliar. At points where communication breaks down because the information is outside the student's psychological field, wrong-answer pages are provided which explain the concept in other terms that can be related to the student's past experience. As each step or concept is incorporated into the student's psychological field, the field becomes extended in this subject area. The program then capitalizes upon this expanded field to develop further expansion.

The level of aspiration principle is used in programming to determine the level of difficulty of the multiple-choice questions in intrinsic programming and to determine step size in linear programming. Success and achievement are measured in terms of level of aspiration. A task that is viewed as very simple or easy is below the student's aspiration level and provides little feeling of success upon completion. A task that is viewed as too difficult is beyond the student's level of aspiration. If such a task is even attempted, there is little feeling of failure when it is not achieved. The ability of intrinsic programs to branch the student into his appropriate level maintains his aspiration to succeed. In order to experience success, the student must always be aware of the possibility of failure.

Linear programming should apply the principles of operant conditioning. The purpose of these principles is to shape behavior in a predetermined manner. This is done by eliciting and intermittently rewarding specified desirable patterns of response. The material is sequenced in small steps that minimize the possibility of error and serve one or more of the following functions: lead-in, augmenting, inter-locking, rote review, restated review, delayed review, fading, generalizing, specifying, dovetailing, or subject matter classification. By presenting each concept under a variety of circumstances and in a variety of ways, the behavior is reinforced.

Gestalt Closure is achieved through skillful review and use of lesson tests. In various fields of endeavor, individuals successfully achieve both short range and intermediate goals. Achievement results in a sense of completion known as closure. In intrinsic programming, the successful answer to a multiple-choice question provides an immediate and limited closure. Confirmation of the right answer accomplishes the same function in the frames of a linear program. However, an endless succession of immediate closures loses its effectiveness in learning situations. To combat this, periodic intermediate closures must also be provided; lesson review and testing can accomplish this end only when they are expertly prepared. A high failure rate on a test fails to achieve closure on the part of the student and a very easy test fails to maintain the student's level of aspiration.

In the preceding paragraphs we have taken a brief look at the interactive effect of various psychological principles that are applied in preparing programmed learning. These are only principles and frequently operate in opposition to one another. In such cases, it is the wisdom of the programmer in selecting his strategy that determines success or failure in the application of the principles.

In the final analysis, the most important contribution of learning theory and psychological principles to programmed learning is in helping the programmer to avoid disastrous errors in judgment. In the preparation of programs, very few programmers consult their psychological principles the way they do their dictionary. The principle of using the question to test the effectiveness of communication and then providing what appears to be the desirable and remedial material is still the most widely used strategy in programming.

Ways to Build the Program

For the purposes of this discussion, the basic strategies used in building an intrinsic program will be discussed in reference to (1) the organization of the material and (2) the organization of the learning process. The organization of the material will be dealt with first.

Under the topic, "Outlining the Program," the step of putting the material into a logical sequence was discussed. It was stated that the term "logical sequence" refers to a frame of reference that is logical to the student. As a general principle, this means moving from the familiar to the unfamiliar. However, moving from the familiar to the unfamiliar does not always involve moving from the beginning to the end.

One very effective method of programming is to move from the end to the beginning. Where the subject area is totally unfamiliar to the student, it is frequently advisable to begin with the final objective or the last skill required in performing the function. The program then moves, step by step, progressively towards the beginning of the function or task. This approach is particularly recommended in teaching skills and problem solving behavior that requires the progressive structuring of nebulous or unorganized material into a well-structured pattern.

For instance, assume that the program was designed to develop decision-making skill in some nebulous area such as assigning priorities to research and development projects. The decision, we will assume, is based upon a compilation of reports and charts indicating the interrelationship of such diverse factors as application requirements, time requirements, budget requirements, manpower resources, special talent resources, and operational policies of the organization.

In such a program, it may be desirable to introduce the student to the final summary report or chart which is the basis for the end decision. Although the student does not know how the various elements of the chart were developed, he is able to use the data for making the correct decision.

In this way, the student gets an overall picture of what it is he is attempting to learn, he enjoys an initial success experience, and he will probably approach the subject with a positive attitude concerning his own abilities. This system has the advantage of keeping before the student all of the elements of the final product, although attention is focused on only one element or step at a time. The student is continually presented with charts, records and materials that are completed up to the task that the student is learning at that given point. As each successive step moving toward the beginning of the function is completed, the material in the following step contains that much less structured information.

This is a "weaning" process which finally results in the student's being able to take a problem without any supportive information or help and carry it through to completion. This approach has the additional advantage of giving the student repeated practice in the skills as he completes each new task or problem up to the point of new instruction.

The other and more common approach to the organizational strategy of programming consists of building the program in the same sequence as the order in which the tasks to be learned are actually performed. This approach is most useful in situations where the order or sequence of learning is as important as the individual tasks. For instance, if you want a child to learn to say the alphabet in its proper sequence, it is probably more efficient and logical to the child to teach it in its normal sequence.

Of course, practically all tasks or job functions have some normal sequence. This is sometimes used by programmers as a rationalization for a conventional organization or approach. The simple fact that a job function is performed in some particular sequence is not always justification for teaching in that sequence. Rather, the programmer should weigh the importance of knowing the sequence and the difficulty of learning the sequence against the advantages of organizing the material in reverse sequence or out of normal sequence.

The second basic approach to building a program is concerned with the organization of learning activities. Basically, information can be taught deductively, inductively, or by a combination of these two approaches. Most intrinsic programs are organized using the deductive approach, and purely inductive programming has, to date, proven to be uneconomical and inefficient.

The completely deductive program is one in which both the total organization of the content and the presentation of the teaching points within each frame are deductively organized. That is, the programmer states at the beginning of the course what the student will learn and how he will learn it. Throughout the course, each new principle or rule is defined or clearly stated and then applications of the rule are discussed. Each page or frame of the program follows the same format of a positive statement followed by application or drill. Thus, the student always knows what he is learning, why he is learning it and how it is applied. This approach is economical because the material is presented concisely and in a straightforward manner. It is supportive to the student because it leads him to successful application of the concept with minimal opportunity for error or misunderstanding. It is the way most subjects and skills are taught and therefore familiar to most students.

The completely deductive program is applicable to any subject involving the teaching of a skill or the comprehension of a body of knowledge. It is less effective when the purpose of the program is to develop creativity or original thinking. Such activity is usually the result of an inductive reasoning process and should, therefore, be taught in an inductive way.

In specialized programs where the objective is to develop skill in creative problem solving, the combination inductive-deductive approach may be more effective. The organization here is to build an inductive program using the deductive approach on each page. In other words, the student may be led to the generalization or the formulation of principles or rules by analyzing the common elements of a variety of situations or applications, but the organization of each individual frame or page will follow the format of a positive statement followed by an application or a problem that involves the manipulation of the concept. It is necessary to use the deductive approach on the individual page format in order to provide the student with the information necessary to solve the problem. Failure to do this results in questions that require "answer guessing" on the part of the student. Answer guessing leads to extremely high error rates, frustration on the part of the student, and a feeling on the part of the student that the programmer is "not playing fair."

The advantage of the combination inductive-deductive approach, in the specialized situations that require it, is that the student is gently guided away from the traditional deductive approach to thinking, and into situations requiring the higher level of inductive reasoning. It permits the use of the multiple-choice question for the purpose of program structure and diagnostic evaluation, and leads the student to develop skill in abstract reasoning and generalization. The disadvantages are that this approach to programming generally requires a longer program to cover a given content area and a high level of programming skill. It also requires careful evaluation and field testing because the final objectives are generally open to greater misinterpretation than those met with the traditional deductive program format. It is a special-purpose technique generally limited to use in extremely high-skill level programs.

The completely inductive appraoch has not proven very satisfactory in achieving measurable objectives through the means of programmed instruction. Although the approach seems theoretically sound, it is very difficult to translate into a written program. Generally such programs give the impression of being formless and pointless. A completely inductive program requires the student to make a generalization based on the information given on each page. This becomes a practical problem because it is seldom possible to provide enough practical situations on a single page to allow an accurate generalization. Frequently, the result is that a number of pages of text are given before enough information is available to ask a question. Then, the question must be so ambiguous that it loses all diagnostic value. This means that the programmer has no way of knowing why the student failed to draw the correct principle from the data provided. Also, it generally leaves the student with the feeling that he is being required to guess.

Since the deductive principle is used in most of our education and training situations, the inductive approach seems foreign, awkward, and difficult to most students. Because of this background, relatively few students are capable of abstracting principles from inductive material. For these reasons, a purely inductive program is generally not recommended as a satisfactory approach for the organization of learning activities and programmed instruction.

Uses of the Student Response - by Norman Crowder

It is characteristic of all teaching machine methods that they require, or at least provide an opportunity for, an active response on the part of the student at frequent intervals throughout the program of instruction. There have been two schools of thought as to the primary reason for eliciting the student's response, which represent historically independent developments and spring from differing theoretical backgrounds.

One school of thought, having its roots in classical experimental psychology, views the student's response as an integral part of the learning process, and therefore as a legitimate end in itself. The adherents of this school are satisfied, in general, to make no further direct use of the student's response than for its assumed effect on the student. Since the student's response is not used to control the program, the resulting programs are of the linear, or non-responsive type.

The other school, having its roots in differential psychology, is primarily interested in utilizing the student's responses to control the course of the programmed material presented to that particular student. Adherents of this second school have generally preferred to beg the question of exactly how the learning takes place, in order to focus attention on being able to determine, from step to step in the program, whether learning has taken place. Given this information in usable form, it is then possible to arrange that the program be automatically modified until the desired result, for that student at that time is attained. This second school is therefore interested in the student's response as the primary datum required to operate a branching, or responsive, program.

It is no more reasonable to ignore the effect of the student's response on the student, than it is to ignore the possible use of the response in automatically controlling and modifying the instructional program. Nevertheless, it should be pointed out that when we consider the use of computing apparatus to modify the instructional program for an individual student, we are primarily interested in this second use of the student's response, and by definition we are contemplating branching, or responsive, programs. In order to make full use of whatever computing capability we employ, we need to realize that a branching program has definitely different structural characteristics from a linear program, and also has different capabilities.

Branching, or responsive, programs utilize multiple-choice rather than completion questions, for a purely practical reason. It is simple, easy, and convenient to let the teaching machine, whatever form it takes, know which of several alternatives a student has chosen and to cause it to take appropriate action on the basis of this choice. It is virtually impossible, in the present state of the art, to let a teaching machine "know" what answer a student has given when the student has written out or "constructed" his answer.

There are some curious cases where there is no difference between the multiple-choice and the constructed response format. These are those cases where the universe of sensible responses is strictly limited to a finite number. Thus, there are only 26 possible alternatives for the next letter in a spelling problem, ten alternatives for the next digit in a problem in decimal arithmetic, and only two in problems in binary arithmetic. In these cases, the distinction between the multiple-choice and constructed response format disappears.

The second major point about branching, or responsive, programs is that when we focus attention on the student's response as the primary datum needed to operate our branching program, rather than as a part of the learning process as such, we become aware that the questions in our program may serve a variety of different functions, and that these different functions require different types of questions. A routine question on a routine step in the program should serve to:

a. Determine whether the student has learned the material just presented;
b. Select appropriate corrective material if the student has not learned;
c. Provide desirable practice with the concept involved;
d. Keep the student actively working at the material; and,
e. Presumably, if the student gets the question right, serve a desirable motivational purpose.

It is quite possible, however, that in writing branching programs we will write questions that serve none of these purposes. We may want a very stiff question, or a short series of stiff questions, to determine whether a student should skip a whole block of material. Such a question, as presented to the student, may not accomplish any of the purposes served by the question used in a routine program step. Again, this point may seem obvious, but there is no reason to view each program step and the associated question as having the same function, and therefore the same structural and statistical properties. For example, it is desirable that on a routine program step (if there is such a thing) no more than 15 percent of the students should select a wrong answer. However, a major program branch might have a question that would be failed by 90 percent of the students.

It may be somewhat less obvious that the alternatives provided in a single multiple-choice question may also serve different purposes and have different consequences. One alternative may be provided to catch a particular procedural error and lead to a single corrective presentation; a second may be such that it appropriately leads to a correctional sub-sequence, while a third alternative may catch an error of interpretation on a point made previously and lead the student back to that point in the program to work his way up again. All of these kinds of contingencies are easy to provide for.

After this unconscionably long preamble about the rationale of responsive programs, we turn now to the major topic: the several means of accomplishing branching programming. We may distinguish two basic types of branching programs. There is the type of branching program in which there is a one-to-one correspondence between the student's answer choice and the next material he sees. Thus, the student's answer choice may lead him to turn to a particular page in a scrambled book or to press a particular button on a machine, at which point a particular, predetermined presentation appears on the viewing screen of the machine. This type of responsive programming requires no intermediate computation between the student's answer choice and the decision as to what material he will see next. This type of responsive programming is called "intrinsic programming." When, in addition to the student's response alone, we use other data and perform further computation to select the next material the student should see, we have what is called "extrinsic programming."

The intrinsic programming technique has the obvious advantage that no complex computing equipment is required for its implementation. The simplest device utilizing intrinsically programmed materials is the "scrambled book" or TutorText, and with relatively simple devices (simple when compared to the complex computers now being proposed) quite complex kinds of branching programs can be achieved.

In principle, it is true that with the simplest form of intrinsically programmed material, the scrambled book, any branching program that depends only on the history of the student's choices can be accommodated, as should be obvious from figure 7.

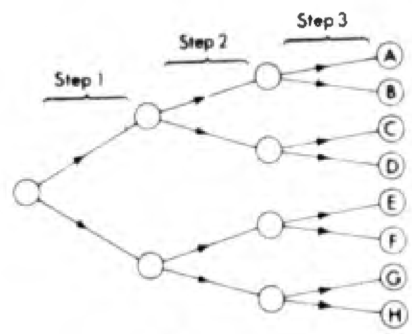


Figure 7

Three-Step Branching Program

In figure 7, a three-step program is shown, with two choices at each step. Each of the students arriving at a different point (A through H) on the third step has a different history, and yet in no step have we considered any data beyond: (a) the student's present location, and (b) his immediate choice.

It would appear from figure 7 that the number of pages or locations to be provided becomes very large if the program is more than a few steps long. In practice, however, this is not the case, since most of the student's history is probably irrelevant for the choice of what to do with him at any given point in the program. If we carry this idea to its extreme, we get the kind of pattern shown in figure 8, which is the simplest kind of branching pattern.

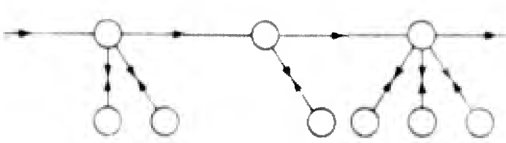


Figure 8

Simplest Branching Program

In this pattern, branches representing errors are "cut off" and the student, after correction, is returned to the point at which he branched, as implied by the double arrows on the branches. With this type of sequence, the number of pages becomes manageable.

A more complex but still practically useful sequence[1] is shown in figure 9.

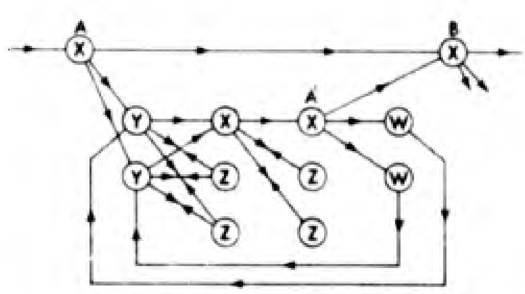


Figure 9

Complex Branching Sequence

[1]The letters W, X, Y and Z in the circles refer to the film codes used on the frames. These codes will be described later in the report.

The chief characteristics of the sequence shown in figure 9 are that a student making any error on Step A is placed in a sequence which will lead him eventually to the next step, but only if he passes the criterion question, which covers essentially the same material as the question A. Again, notice that in this sequence the only data required to determine where the student should go next is his present location and his immediate choice. The branching program can therefore be accommodated in such a simple device as a scrambled book.

A somewhat more complex device, the AutoTutor Mark II shown in figure 10, handles similar material, but adds some new dimensions of control. The actual material is handled on a roll of 35mm microfilm. The student sees this material projected on a view screen, and has ten response buttons which he uses to answer the questions.

The actual effect of the ten buttons is as follows:

Button I	-	rewinds film 19 steps
Button H	-	advances film 15 steps
Button G	-	advances film 13 steps
Button F	-	advances film 11 steps
Button E	-	advances film 9 steps
Button D	-	advances film 7 steps
Button C	-	advances film 5 steps
Button B	-	advances film 3 steps
Button A	-	advances film 1 step
Button R	-	cancels previous excursion

Figure 10

AutoTutor Mark II Teaching Machine

The reason that the film advances an odd number of steps on each excursion is to avoid interference between successive questions. Thus, if question N appears on film frame 106, for example, and the answer choices are buttons A, B, and C, then the material for the three responses will appear on frames 107, 109, and 111, respectively. Let us say that A is the correct answer to question N. Then frame 107, reached by pressing button A from 106, will now contain question (N + 1).

Again, let the answer choices for the new question be A, B, and C. The material for these responses will appear on frames 108, 110, and 112 respectively. These frames are available because of the odd number of steps taken by the film advance mechanism. Thus, interference between successive questions (first order interference) has been avoided. However, we must now be more careful. If we let A be the right answer to question (N + 1), then the next question (N + 2) will appear on frame 108. For question (N + 2), we cannot then use buttons A and B for responses, since the corresponding response frames (109 and 111) are occupied by the B and C responses to question N (frame 106). In such a case, we would simply use buttons C, D, and E (leading to frames 113, 115, and 117 respectively) for the choices offered to question (N + 2) on frame 108.

We have assumed the worst possible case here (two successive A answers correct), of course, to demonstrate what we mean by interference between questions. Interference between successive questions never occurs, because of the odd spacing of response frames, but second order interference (between one question and the second succeeding question) is possible, and would be handled by choosing the higher lettered response buttons, as was shown.

Buttons A through H are the ones normally used by the student to answer multiple-choice questions. Each button advances the film the number of steps shown above figure 10. If the student has chosen the correct answer to a question, the film frame which is then shown him will contain new information and a new question. When the student again presses one of the A through H buttons to answer the new question, the film will advance from the new location. If the student has chosen an incorrect answer, the material he sees will, in the simplest case, present correctional material and then direct him to press the R or RETURN button, which will reverse the excursion the machine has taken immediately previously, and put the student back on the original question frame, to have another try at the question. This is the sequence of events, somewhat more conveniently arranged, that would occur if the student were using a scrambled book. However, as we have the material inside a machine, under mechanical control, we are able to add some very interesting new control features to the material. These controls are brought about by a code on each film frame, which determines which buttons on the machine will be allowed to operate when the student is viewing the frame. There are four codes on the Mark II AutoTutor, with the following functions:

Code X — All buttons operate. This is the code used routinely on a right answer frame. The student has just chosen the correct answer to the previous question; we are satisfied with his performance and have no reason to restrict the student's activity, nor does the student have any reason to do anything but what we want him to do, which is to choose an answer to the new question. If the student should want to review the previous right-answer frame, he may do so by pressing the RETURN button, which we allow to operate.

Code Z — Only the RETURN button works. All other buttons are dead. This code is normally used on a simple wrong-answer frame. The student on such a frame has made an error, and we want him to read the correctional material and then to return to the question page. We arrange the situation so that only the RETURN button works at this point.

Code Y — Only buttons A through H work. This code is used on a wrong answer frame that introduces a correctional sub-sequence. The student has made an error; we want him to work forward through a correctional sub-sequence. In particular, we do not want him to return to the question to try another answer. Therefore, we do not allow the RETURN button to work from this frame.

Code W — Only the I button works. The I button on the Mark II is the only button that runs the film backwards. The I button, with the W code, is used to run the student back in the program, to review material previously covered, or, in a sequence of backward excursions, to bring the student back to the beginning of a complete section of the program.

With the use of these film codes, which appear on each frame of instructional material and are read by two photocells in the machine, it is possible to achieve great flexibility and at the same time exercise complete control over the options available to the student. It is literally impossible for the student to advance from frame 1 to, say, frame 400 of a program until he achieves the criterion performances we have required on the way. The use of these codes in a sub-sequence is illustrated in figure 9, where the appropriate codes are marked in the circles representing frames.

It is a nice point whether the Mark II AutoTutor, with the film codes described, uses intrinsic or extrinsic programming. Whatever conclusion we might come to on this question of definition, it is nonetheless true that we use only data derived from the student's present position, the film frame he is presently viewing, and his immediate choice to determine what will happen to him next. The point in presenting this information in this much detail is to give some idea of the complexity of programming that can be accommodated without involving the more complex types of computing apparatus. Actually, sequences with a much greater degree of complexity can be developed with the same techniques, and are commonly used.

How and When to Use Sub-sequences

Sub-sequences are, in a sense, a specialized use of the student response for remedial or enrichment purposes. Therefore, everything that has been stated about student response control applies to the programming of sub-sequences.

A sub-sequence is a series or sequence of right- and wrong-answer pages or frames inserted into the program for a special purpose and intended for only a limited portion of the total student population. Most sub-sequences serve a remedial purpose. That is, they provide instruction in an area that does not contribute directly to the program's objectives, but that is essential to the completion of the program.

Generally, the subject matter of the sub-sequence falls within the category of prerequisites to the course that the student is assumed to have. However. if it is recognized that review of certain material may be required by a significant portion of the students taking the program, then it may be decided to include the material as a sub-sequence. The inclusion of such material broadens the program's usefulness and adapts it to a wider range of individual student needs.

It is generally not desirable or practical to attempt to teach fully or cover a course prerequisite in a sub-sequence. In cases where the student does not possess the prerequisite information, he should be instructed, either in the program or by the person responsible for training, to prepare himself with a program on the particular subject he needs.

The sub-sequences, then, are ordinarily used to review a portion of a prerequisite or to deal with some special application of the knowledge. For economic reasons, the sub-sequences should usually be fairly short.

Another type of sub-sequence, less frequently used but equally valuable, provides enrichment material. In a program designed to teach how to perform certain functions, it may sometimes be desirable to provide information concerning the "whys" of the subject for the intellectually curious. Of course, this is frequently done to a limited degree within the framework of the "right-answer" stream. However, the extremely bright student who likes to work with abstract concepts may need deeper stimulation to keep him motivated. The determination of whether to include or exclude such material is usually based on economic considerations and training time factors. If these permit, such sub-sequences are often desirable.

The student may enter a sub-sequence automatically through the selection of a particular answer alternative or through the provision of an option written into the program. The option will usually be an alternative stated in such terms as, "If you wish to know why this formula is used, turn to page X (or press button D)."

Remedial sub-sequences may also be entered at the student's option. In such a case, the instructions might be, "If you wish to review the Rule of Decimals, turn to page X (or press button D)." When an optional review is provided in a program, it should usually be followed, in another frame, by a diagnostic question in which one of the alternatives automatically branches the student into the sub-sequence. In this way, the student who needs the review but does not elect the option is identified and branched automatically.

The correct choice on the last alternative of a sub-sequence should always lead back into the mainstream of the program at the appropriate point for that student. In most cases, this will be the point in the program at which the student left the mainstream. The only exceptions would be in cases where some of the material in the mainstream has, of necessity, been covered in the sub-sequence. In such a case, the student would return to the frame following this material, and the programmer must be certain that the right answer to the last sub-sequence frame is the same as the right answer to the question in the mainstream which leads to the same frame.

There are two other types of branching which are not technically sub-sequences but possess many of the characteristics of the sub-sequences and will therefore be discussed under this heading. One is multiple-tracking and the other is express-tracking.

Multiple-tracking consists of a series of mainstreams dealing with the same subject at various levels of difficulty or abstraction. These tracks may or may not make provision for the student to move from one level to another. If such provision is not a requirement of the program, then it is less expensive and more practical to prepare separate programs at different levels of difficulty for handling students of different abilities.

Students can be assigned to the various levels of the program on the basis of a pre-test. It is only in cases where it is necessary to allow the student to move from one level of complexity to another that multiple programming is economically feasible.

Express-tracking consists of difficult diagnostic tests built into the program for the purpose of determining the student's ability to bypass certain topics essential to the program's objectives. Generally speaking, a single question should not be used as the only evidence of the student's knowledge. In practice, the procedure usually consists of either a diagnostic question or an optional question for the initial separation of the students. Those students who elect the express track option or who answer the diagnostic question correctly are then usually given one or two additional questions (one question per frame) covering the topic in further detail. Those who fail these additional questions are dropped back into the mainstream of the program and those who answer successfully are sent ahead to the next topic in the program. This procedure is very effective in reducing study time in situations where the student population is heterogeneous.

Economic considerations are always an important factor in planning and using sub-sequences. Only a portion of the student population will ever see the sub-sequences. If this proportion is too small the sub-sequence may not justify the cost of its inclusion. If a sub-sequence is used by much more than 60% of the student population, it may indicate that the assumptions concerning the student's prerequisite knowledge were faulty or that the material should be considered as a part of the mainstream.

This type of information can be determined from the field test results. At this point, the dollar cost of including the sub-sequences can be accurately evaluated against the projected man-hours of study that the sub-sequences will provide for the anticipated student population.

Constructing Multiple-Choice Questions

The primary purpose of the multiple-choice question in intrinsic programming is to ensure understanding of each unit of information at the time that it is presented and before the student moves on. This implies a diagnostic function of the question and a remedial application of the results.

There are three major diagnostic uses of the multiple-choice question and the answer alternatives. They are used to diagnose a remedial need for basic or background information; they are used to diagnose a failure in communication or a misunderstanding of the concept; and they are also used to detect the student who is skimming or guessing. These three uses account for about ninety-five percent of the frame or page questions used in intrinsic programming. The other five percent are used for special purposes, such as separating students on the basis of their abilities to think at different abstract levels or for providing enrichment, express tracking, or optionally selected material.

Not every page or frame of an intrinsic program will perform all of the major diagnostic functions. Obviously, only a frame which is intended to lead into a sub-sequence will have an answer alternative designed to diagnose this need. However, every frame should have at least one alternative intended to test the communication of the material and to determine the nature of the student's misunderstanding. Alternatives should be frequently provided to detect the student who is skimming or guessing. The alternatives should always be written so that they are challenging at the level of the intended student population.

An alternative is challenging when it is difficult enough to present the possibility of failure yet simple enough to inspire the student to tackle the problem. Most students do not attempt the impossible. If a problem seems completely beyond their grasp, they give up without trying and without experiencing a feeling of failure. Such a problem is above the student's "level of aspiration." Consistent use of such problems in a program is undesirable because it frustrates the students and results in high attrition rates. On the other hand, a student feels no success over the accomplishment of a task that "anybody could do." To really experience success, the student must feel that there is a real possibility of failure any time he doesn't give his best to the selection of the answer alternative.

A question that is too easy is said to be below the student's level of aspiration and a question that is too difficult is said to be above his level of aspiration. Most questions should fall within the range set by these two limits. However, maintaining a consistent level for all the questions of a program is not desirable, because the questions would probably then be too difficult for the students at the beginning of the course and too easy at the end of the course. In this sense, the level must rise continually as learning takes place and builds upon previous learning. However, even the rise should not be consistent. Within the range of the student's level of aspiration, the questions should fluctuate between the upper and lower limits in an inconsistent or undiscernible pattern. For instance, two or three relatively easy questions might be followed by a rather stiff question constructed close to the student's upper level of aspiration, followed by another easy question and then perhaps two moderately stiff questions. This undiscernible pattern keeps the student on his toes with fairly tough questions and reassures him of his continued progress in situations where the questions are less difficult.

How are multiple-choice questions constructed to serve this function? First of all, the programmer must know his students, their capabilities, and their aspiration limits. Beyond this, there are certain helpful guides or rules to be used in the framing of the question. We will now examine some of these rules.

The multiple-choice questions consist of two parts: a stem or lead and a list of alternatives, at least one of which is correct or clearly the best choice. The stem may be written in various forms. However, the direct question and the incomplete statement are the two most widely used. Examples of these are shown below.

The Incomplete Statement
Example: The primary colors:
red, green, and yellow
red, blue, and yellow
red, orange, and blue

The Direct Question
Example: What are the primary colors?
red, green, and yellow
red, blue, and yellow
red, orange, and blue

Of these two forms, the direct question is generally preferred. Experience indicates that the direct question is more effective and stimulating to the decision-making process and is less apt to lead the student into answer guessing. The question should contain at least three alternatives. In rare instances, only two alternatives may be logical; however, the use of only two alternatives increases the likelihood of guessing since the student has a fifty-fifty chance of being right.

The alternatives may be in any form that will fulfill the logical requirements of the question. They may be sentences, phrases, clauses, single words, letters, formulas, diagrams, pictures, or numbers. Whatever the form, the alternatives should require the student to participate. He must think about the information given and he must use the information to solve a problem, make a judgment, or offer an interpretation. The alternatives should not merely require the student to recognize a phrase that has been used in the expository text.

The following rules will be found helpful in writing multiple-choice questions for intrinsic programs.

Rule 1. Test the student on the central issue, not on a trivial secondary point.
Rule 2. Never attempt to deceive the student; avoid hinging the correct answer on an "always," "never," or "possibly."
Rule 3. Write questions which require the student to use the information you have given him.
Rule 4. Write questions which any student who has the necessary background could answer. (Make questions neither too easy nor too hard).
Rule 5. Never make the student guess at an answer.
Rule 6. Use the following alternatives sparingly:

I don't know.	None of the above.	Yes.
I don't understand.	All of the above.	No.

The student should never be required to guess at an answer. All the evidence should be placed before him. There is a great temptation to construct multiple-choice questions leading to the next logical step in the development of the subject, and let the student guess his way through the course instead of working his way through. The program writer who has labored diligently to explain a certain point may reason that it is now perfectly understandable, and to ask a question on what is so clear and self-evident seems quite redundant. But, this apparent redundancy is an important element of the intrinsic programming method.

The program writer should also avoid questions centering around some trivial point. The object of the program is to teach the student - not to outwit him. It is fairly simple to construct a question in which the correct answer depends on an "always," a "never," or a "possibly," when the student cannot reasonably be expected to know which is correct; or to avoid the central issue entirely and question the student on some peripheral statement which lends itself readily to construction of a question. However, the student will understandably resent such subterfuge.

It is vital that the student feel that the program writer is "playing fair" with him. The first thing that a student learns from a properly constructed program is that he can learn if he will pay attention. It is disastrous to turn an intrinsic program into a guessing game.

Another pitfall in the construction of the multiple-choice question is the temptation to ask a question which does not require thought but merely the ability to look back at the text and find the answer. It is admittedly hard work to construct a question that will really test the student's ability to use new knowledge.

The following principles will be found helpful in constructing multiple-choice questions that comply with the rules enumerated above.

Principle 1.	Test only one concept at a time. (FOCUS on the idea you want to communicate.)
Principle 2.	In the stem of the question, include all words which are common to all of the alternatives.
Principle 3.	State the question in positive terms. Do not ask questions such as: Which of the following statements is false? Which of the following is not an example... ? Which of the following statements is not false?
Principle 4.	Make the alternatives logically consistent with the stem of the question.
Principle 5.	Make the alternatives grammatically consistent with the stem of the question.
Principle 6.	Make the alternatives similar in grammar, length, content, and degree of precision.
Principle 7.	Make the alternatives as brief as possible.
Principle 8.	Make the alternatives plausible.

Construction and Use of Lesson Tests

The rules and principles for constructing multiple-choice questions for intrinsic programmed pages apply equally to the construction of multiple-choice questions for lesson tests. A lesson test follows a chapter or unit and may serve one or more of several functions.

The lesson test should always be designed to evaluate the student's comprehension of the material covered in that unit. Generally, it is not feasible nor desirable to retest every single concept presented in the lesson. Therefore, the lesson tests are usually broader and more general in nature than the intrinsically programmed pages. There may be as few as three questions but they will be broad in scope and will require the student to demonstrate an integrated understanding of the subject and the ability to perform the required functions.

A diagnostic function is frequently added to the lesson tests, especially when the AutoTutor Mark II is used and strict controls can be exercised over the student. When the test is to perform this diagnostic function, the test and the program are prepared so that certain of the test questions cover certain areas of the programmed unit. A predetermined criterion of performance is established for each of these sections or units, and the TutorFilm is then prepared so that failure to meet the specified criteria will automatically require the student to review the topics in which he has shown a weakness.

This remedial review can be provided in two ways. The most common form is the "washback" which means that the student is returned automatically to the section of the program that he is required to restudy. In this case, the student must then work forward from that point on to the end of the lesson. The second trip through the lesson is, of course, more rapid than the first one, but nevertheless this can be a time-consuming process if the lesson is a long one. In most cases, the washback is a frustrating experience to the student and tends to build up resentment toward the program.

Another method of accomplishing the same purpose is to provide the review material as a sub-sequence and to return the student immediately to the question after the section designated for review has been completed. To reduce the production costs of this method of compulsory review, the same expository text can be used but fitted with different multiple-choice questions and problems. The mechanics of planning and arranging this type of review and the washback are discussed under the topic, "How to Scramble For The AutoTutor Mark II."

A third major function frequently performed by the lesson test is to provide the student with a self-appraisal of his progress. Care must be taken in preparing programmed instructional materials to prevent the student from feeling that he is lost in a forest of isolated items of information. The tendency toward this feeling is greatest when the objectives of the program are not clear to the student, and when he has no opportunity to take a broad look backward over the territory he has covered. The lesson test will fill this last need. To do this effectively, the lesson tests should not test the same isolated bits of information as the programmed pages, but rather should lead the student into seeing the synthesis of the parts.

Although the construction of lesson test items is more critical than the individual frame items and the questions must be broader and also more comprehensive, there is no different or unique skill required for constructing lesson test items. The principal differences are in the format of the test items and in the uses made of the resulting information. The format of the test question is usually determined by the programmer's purpose and the medium to be used ultimately in programmed presentation. For instance, the control features of the AutoTutor Mark II open up some unique uses for the lesson tests which are not available in any other mode of presentation.

In the TutorText book format, the lesson tests are usually handled in one of two ways. The most elaborate places the test question for each lesson or chapter in an appendix at the end of the book. The student is told to work through each problem of the lesson test and then to turn to a specific page of the book to check his answers. He is also told that if his answer is incorrect, the explanation of the point he missed is to be found in the textual matter on that page.

A simpler but less effective method places a short quiz at the end of each chapter. Each question is assigned a letter: F, D, A, C, E..... Answers are found on the opposite page in alphabetical order: A, B, C, D, E, F. This device merely ensures that in checking the answer to one question, the student does not accidentally glimpse the answer to the next.

More complex lesson test sequences can be used in the AutoTutor Mark II. The function of these, as has been pointed out, is not only to consolidate gains and provide practice, but also to provide additional remedial material or drill for those students who indicate a need by failing to pass a predetermined criterion for success. The technique of providing this control over the student's progress is discussed under the section on scrambling. It is important here, however, to state briefly the principles upon which the control techniques are based.

The AutoTutor Mark II has a theoretical capability of branching or separating students into as many as nineteen different tracks. In practice, this potential is never achieved nor is it necessary or desirable. However, it is the basis for the selective screening accomplished by the lesson tests.

This is the way the principle is put to use. First of all, the lesson is divided into subsections or units that can be covered by a single broad question. Next, a series of questions is constructed for each of these sub-units or sections of the lesson. Usually, three parallel questions for each section are sufficient. The questions are then given identification numbers which refer to the sub-sections the questions cover. For the purposes of explanation, assume that the lesson contains the three sections X, Y, and Z and that three questions have been constructed to cover each section. Thus, the lesson tests will consist of nine questions: X-1, 2, 3; Y-1, 2, 3, and Z-1, 2, 3.

If the washback system of review is to be used, the test items Z-1, 2, 3 will be placed first in the lesson test, followed by items Y-1, 2, 3 and X-1, 2, 3. For mechanical reasons that will become obvious with understanding of the scrambling process, this arrangement of the questions makes it possible to carry the student who misses only the last section (Z) back to the beginning of that section for review. The student who misses questions dealing with the first section (X) however, will be sent back to review the entire unit.

If the selective sub-sequence method of control is used, the programmer has greater flexibility in the arrangement of the test items. In this system, the test items are usually itemized. For example, the order of the questions might be Z-1, X-1, Y-1, Z-2, X-2, Y-2, Z-3, X-3, and Y-3. Any order or combination of the test items is possible with this system.

A predetermined criterion for passing is established. For purposes of our discussion, let us say that the criterion will be a minimum of two correct answers out of the three items on each section.

All the students will approach the test at the same point. In our example, that would be item Z-1. Assume then that all the items have three alternatives, one of which is correct. From item Z-1, the program develops two branches. Those students who answer the item correctly will approach item X-1 with no errors against them. The students who choose either of the two wrong answers will approach item X-1 on a different frame with one error credited against them. Question X-1 must be on a different frame for these two groups of students for two reasons. First, this is the only way that the machine is able to maintain a record of their pass and fail performance. Second, the students who answered correctly are told that their answer was correct on the frame that presents the next question (X-1) to them. The students who answered the first item incorrectly are told that they now have one error against them.

From the second test item, four separate tracks would emerge: two from the right-answer track of question one, and two from the wrong-answer track of question one. This process of branching is continued throughout the test. When the student misses two questions in reference to a given sub-section (for example, Z), the answer choice carries the student to the first frame of the sub-sequence reviewing that section. The last frame of the sub-sequence can bring the student either to the beginning of the test or back to the point at which the student left the test. This is an option to be determined by the programmer.

PAGE FORMAT AND MANUSCRIPT ASSEMBLY

There are two basic page or frame formats in an intrinsic program: the first is the format of the right-answer page and the second is the format of the wrong-answer page. The first page of the program and the lesson tests are variations on the right- and wrong-answer pages. These four types of format will be dealt with in this section. The title page, table of contents, and appendix pages follow standard book practice and so will not be discussed here.

Right-Answer Page

The basic format of the right-answer page is identical for both book presentation and AutoTutor Mark II film presentation with a few minor exceptions involving terminology. In machine format, the term "frame" replaces the term "page" used in book format to refer to a step in the program. In book format, the student is instructed to turn to a given page number. In TutorFilm format, the student is instructed to press button ___. Although the final copy of a TutorFilm is typed on a special animation paper, the other elements of the format remain the same.

The first element of the format for the right-answer page or frame is the identification number. This appears in the upper right-hand corner in both instances. In book format, it is called the page number and in film format, the frame number. In book format, it is desirable, but not essential, to have below the page number a parenthetical statement "(from page ___)." This keeps the student from getting lost in the program. For purposes of reference, a sample right-answer page is shown as figure 11. The elements of the format have been identified on the sample by means of alphabetical reference. The page number is identified by reference (A).

(B) The first paragraph of the right-answer page consists of a restatement of the student's answer choice. The words YOUR ANSWER: should appear in caps. Following this should be a complete statement consisting of enough of the question stem plus the selected alternative to enable the student to recognize the content of the question. This is provided as a reference for the student. Fast students and students doing well in the program frequently omit reading this paragraph. This procedure is permissible since it speeds up progress through the program, but the necessity of providing this information remains.

13
(from page 7)

(A)

(B) YOUR ANSWER: Self-pacing means that a student's progress through a program is determined by his comprehension of the subject matter.

(C) You are correct.

The program's capability of adapting to a wide range of student needs is a distinctive characteristic. The effective utilization of this capability reflects the skill of the program writer.

(D) This brings us to the fourth characteristic of intrinsic programs. Programming keeps the responsibility for teaching on the writer. If the student is unable to grasp the program content, it is the writer's failure rather than the student's. Textbooks frequently rely upon the instructor to clarify points, insure communication, and evaluate comprehension. The program writer must make certain that these elements are built into the program.

(E) How does the program writer insure communication when preparing an intrinsic program?

a. By constructing questions that will have very low error rates. page 15

b. By constructing questions that diagnose misunderstandings that may occur. page 19

c. By constructing questions that are challenging to the student. page 21

Figure 11

Sample Right-Answer Page

(C) The second paragraph of the right-answer page begins with the statement YOU ARE CORRECT. This statement is followed by one or more sentences that summarize why the chosen answer was correct, comment on the other answer alternatives, and build a bridge between the answer choice and the next new concept. All three of these functions of the second paragraph are not always required, but the first function should always be provided. There is some academic dissension concerning the consistent use of the exact words, "YOU ARE CORRECT" in the second paragraph. Some programmers feel there is an advantage to be gained by varying the phrase used to convey this information. However, there is evidence that after the student has completed a few frames of an intrinsic program, this statement becomes a psychological signal indicating that the student is on the right track rather than a phrase to be read. It appears that after a short conditioning period, the student does not really see the phrase, "YOU ARE CORRECT." as a series of words but rather as a psychological reward for a correct response. It would seem that the signal value of the statement in unvaried form outweighs the possible advantage of providing variety in the phrase. It is, therefore, recommended that this term not be varied.

It has further been observed that fast students frequently omit a detailed reading of both the first and the second paragraph and move directly to the third paragraph. This time-saving procedure is possible only so long as the student can be sure that these two paragraphs will invariably contain only the information that has been discussed. In other words, the first and second paragraphs should never contain new information essential to the conceptual development of the program.

(D) The next element in the format of the right-answer page is the expository text used to develop the next sequential concept in the program. This will usually consist of from one to three paragraphs written in a concise but informal manner. The writer should demonstrate a sensitivity to the needs of the student and recognize where a student may go wrong. He should understand where encouragement, warning, or other clarification may be in order. This skill and willingness to take the trouble to write clearly, effectively, and accurately involves more than impeccable grammar. Although the difference between good and bad writing is not always susceptible to scientific analysis, it can usually be measured in terms of readability and reader interest.

As far as possible, the program writer should attempt to duplicate the personal relationship between a student and his tutor. This does not mean that the writer should strain for a light touch or "talk down" to the student. Rather, whatever the tutor would say to the student in a face-to-face relationship can and usually should be put on paper or film. If it is considered desirable for the student to take notes, he should be told to do so. If something must be memorized, the programmer should so instruct the student. Parenthetical remarks, warnings, encouraging comments, and even occasional flashes of humor are quite acceptable, just as they would be in a live tutoring session. It is essential that the program writer understand, when he is preparing the expository text, just what the student knows at that point, what he is being told that is new, why he is being given this new material, what he is expected to do with it, and what mistakes he might possibly make. The programmer should determine the places in this portion of the text at which the student might be misled, and rewrite the material to eliminate as many of these traps as possible. The possible misinterpretations that remain provide the basis for the multiple-choice question on that page.

(E) The last element in the format of the right-answer page consists of the multiple-choice question and the answer alternatives. Techniques and principles for constructing the question and alternatives have been discussed previously under "Programming Technology." In most situations, it is preferable to use the direct question rather than the incomplete statement. The student should be asked meaningful questions, not quizzed on trivial points. He should not be required to guess at an answer. It is important that the student feel that the programmer is "playing fair" with him, but it is also important that he be required to think, not just to look back at the text and find the answer.

Each of the answer alternatives should be followed by instructions to turn to a stated page or to press an indicated button on the AutoTutor. These instructions can consist merely of the statement "page 15" or the button letter that is to be pressed. For convenience to the student, these instructions should be lined up one below the other on a line with and to the right of the question within the margin of the frame or page.

Wrong-Answer Page

The format of the wrong-answer page is so simple that many of the finer points are overlooked even by experienced program writers. Using figure 12 as an example of the format of the wrong-answer page, its characteristics will be discussed point by point.

(A) The pagination for wrong-answer pages in both the TutorFilm and the book format is the same as for the right-answer pages and should be consistent with the other portions of the program. If the statement "(from page__)" is used on the right-answer pages, it should also be used on the wrong-answer pages.

(B) The first paragraph of the wrong-answer page is handled in exactly the same manner and format as the first paragraph on the right-answer pages. The paragraphs should begin with "YOUR ANSWER," followed by enough of the question stem and the selected alternative to make a complete sensible statement.

(C) The first sentence of the second paragraph should clearly indicate to the student that his selected answer was not correct. There is no standard or required format such as "you are wrong" for providing this information. Rather, it is preferable to vary the wording of this information and to tailor it more specifically to the type of error made by the student. If the standardized statement "YOU ARE CORRECT" is used consistently on the right-answer pages, its absence from the wrong-answer pages will again serve as a signal to the student that his answer was incorrect.

The programmer should do the student the courtesy of assuming that the student made a sincere attempt to understand the material and failed for some good reason. This attitude on the part of the programmer will preclude the use of sarcastic reprimands on the wrong-answer pages. Phrases like "Oh, come now" or "That was a silly answer" have no place in an intrinsic program. The programmer should remember that he constructed the answer alternative and if the alternative is silly or stupid, it is of his own doing. There are occasions, of course, when the programmer will want to be rather straightforward or firm with the student. These occasions usually occur on wrong-answer pages resulting from alternatives intended to catch the student who is skimming or guessing without really studying the program. Even here, the program writer can be firm without being antagonistic or sarcastic. Lists of ways to tell the student that he is wrong have been developed and are available, but they are not recommended. In the final analysis, it is better to tell the student that he is wrong in a way that suits the particular response he has given.

The balance of the second paragraph should summarize briefly why the selected alternative was incorrect. This function is similar to the function of the second paragraph on the right-answer page.

(D) The subsequent paragraphs of the wrong-answer page are devoted to the more detailed elaboration of why the selected alternative was wrong and how the student can proceed to solve the problem correctly. The programmer should maintain the attitude that the student is entitled to a further explanation, and just as some elusive point may be suddenly illuminated by consulting a different author, so a slightly different approach on a wrong-answer page may clear the matter up.

In some cases, it may be necessary or desirable to provide additional information bearing on the topic if it is appropriate to that particular incorrect response. Information that is essential to the conceptual development of the program should never be presented on the wrong-answer page unless it is also provided in the mainstream of the program. Violation of this principle will result in the omission of essential information for the majority of the students who will never see the wrong-answer page.

The information on the wrong-answer page should give the student not the answer, but rather the information necessary to enable the student to solve the problem himself. If the correct answer is given to the student, the possibility of testing this second effort to communicate is lost and the student may proceed in the program carrying with him a misconception. The student's choosing of the correct solution should always be the basis for movement to the next new concept. The only exception to this principle occurs in situations where the question presents only two alternatives. In such cases, the student can arrive at the correct answer merely by the process of elimination. This is the reason that questions with only two alternatives are undesirable.

15
(from page 13)

(A)

(B) YOUR ANSWER: The writer insures communication by constructing questions that will have very low error rates.

(C) A low error rate would not insure communication. The low error rate might merely mean that the questions were very easy or that the correct alternatives were very easy or that the correct alternatives were very obvious.

(D) Actually, some programs are written to preclude the possibility of the student's making mistakes. Such programs rob the student of the opportunity to make decisions and to test his own understanding of the concepts. Since it is assumed that errors will be practically nonexistent, wrong-answer pages are eliminated and every student is forced to read all of the material. Such programs are long and either boring for the bright student or paced too rapidly for the slow student, and a sacrifice of adaptability and self-pacing features results.

As you may have discovered from reading this page, an incorrect choice can be a learning experience if the nature of the error is detected and corrective material provided. You can prove this point by returning to page 16 and examining the other alternatives with this point in mind.

(E) Return to page 16.

Figure 12

Sample Wrong-Answer Page

(E) The final element in the format of the wrong-answer pages should be instructions to the student directing him to the next point in the program. In a TutorFilm, this instruction will consist of directions to press an indicated button. In book format, the instruction will in most cases be to return to the right-answer page from which the selected alternative was derived. If the previous right-answer page has only two alternatives, the student's time will be saved by instructing him to turn directly to the page assigned to the right answer. The exception to this is in cases where the programmer wants the student to re-read the previous right-answer page material.

First Page of Program or Lesson

The first page of the program or lesson should conform to the form of the standard right-answer page except for the following:

The first paragraph of a typical right-answer page restates the previous question and the answer selected by the student. This restatement is, of course, omitted on the first page because there is no previous question. The second paragraph of a typical right-answer page restates the reason why that answer choice was correct and bridges into the next concept. This paragraph is also omitted on the first page for the same reason.

In the place of these two paragraphs, the first page of the program and the first page of each lesson will contain one or more introductory paragraphs used to motivate the student or summarize the objectives of the new unit. In rare instances, this information may require so much space that the actual program format with its multiple-choice question will be moved back one or two pages into the program. This is undesirable and should be avoided whenever possible.

Format of the Sub-Sequence

The first page of the sub-sequence contains elements of both the right- and wrong-answer pages. Elements A, B, C, and D are the same as for a wrong-answer page. Within the format of element D, the student should be told that this is the beginning of a sub-sequence. Also contained in this section of the format will be the first new concept of the remedial material.

Following this expository text, the first page of a sub-sequence will have a multiple-choice question and answer alternatives. These replace the return instructions of a typical wrong-answer page. The format and the construction of the question and its alternatives are the same as for other right-answer pages. The balance of the sub-sequence follows the format discussed for right- and wrong-answer pages.

The final page of a sub-sequence will usually present the same multiple-choice question as the right-answer page that precedes the point of entrance into the mainstream of the sub-sequence. That is, the selection of the correct answer on either the sub-sequence or the mainstream will bring the student to the same page or frame. The wrong answer alternatives of the final page of the sub-sequence may either lead to the same wrong-answer pages for this purpose because it is less confusing to the student.

General Format Information

A good lesson in programmed instruction carries the student along a definite route, with side trips clearly identified. The student must have a strong sense of direction in order to make clear decisions. Even more, the program writer must maintain a similar sense of direction in order to avoid meaningless side trips and extraneous information. This does not mean that additional meaningful and motivating information should be avoided. The criterion for inclusion of additional information is, "Does this information contribute to the overall objectives of the instructional lesson?" Each page must make a contribution to the overall purpose of the lesson and it must also make its own point clearly, or further develop or further review the previous point. In this connection, the following rules may be found helpful.

Rule 1 — Make your point on each page and again in each five to twenty pages, and again in a review.

Rule II — Continuity – The level of difficulty of each question must fall within a range that is respectful and considerate of the student's ability. No question should be absurdly easy. The instruction must move along smoothly, but it should not gloss over inherent ambiguities or difficulties. Be careful not to load on too much new information before reviewing and practicing.

Rule III Watch the new information rate. Every new word and idea must be used several times before the student can be expected to accept it without question. It is helpful to require the student to make a response using the new term. This is equally true with each new formula introduced or each new type of calculation to be made.

Rule IV Avoid sarcasm.

Rule V Divide long sessions into sections with appropriate comments to the student. A lesson should normally contain from 12 to 18 right-answer pages. If it is necessary to have a lesson longer than this, divide it into parts with a review and short test sequence. Then, inform the student simply and directly of his progress at the end of each part and of the nature of the work remaining.

Sub-sequences should be used when the need is indicated by:

1. Lack of retention of material presented earlier in the program.
2. Lack of adequate presumed background (prerequisite) information.
3. Need for additional instruction to develop a thorough understanding of the present topic.

Avoid long sub-sequences. If the topic takes five or more wrong-answer pages, it probably deserves treatment as a separate (perhaps, optional) lesson. It is desirable to provide several entries into sub-sequences. This will minimize the chances of a student's going past a needed sub-sequence by guessing.

It is the program writer's responsibility to indicate on the copy how he wants illustrations placed or equations lined up. It is advisable to use punctuation in mathematical equations sparingly. The addition of running commentary words, like "if," "so," "and," "or," "hence" should be reserved for mature students.

The use of language that refers to the technology or the mechanics of programming should be avoided unless the program deals with this subject. This includes such terms as "wrong-answer page," "I Track," "right-answer stream" or "main line of instruction."

In mathematical programs, numerical answers should be listed in ascending or descending order. In other programs, the answers should be listed in the order the program writer wishes them to appear on the page. Only the writer can determine the best order for the answers to appear in order to avoid giving outside cues to the correct one.

Express tracks or supertracks are a convenient method of reviewing prerequisite material. The procedure and format consists of the following steps:

(1) Divide the topic into its essential concepts;
(2) Provide for two questions to review and test each concept;
(3) Allow the student some measure of self-determination (that is, an option as to taking the supertrack or the full program).

If the supertrack questions are detailed enough, no final testing should be necessary.

Lesson Test Pages

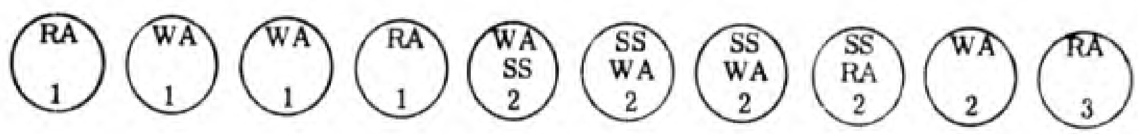
The format of the pages containing lesson tests is completely dependent upon the form of test being used. Thus, the format should be designed around the requirements of the test instead of forcing the test to conform to a rigid format.

Manuscript Assembly and Pagination

The page numbers assigned to a manuscript during its preparation are not the same as the final page numbers assigned to the program when it is in scrambled format. Since the scrambling process comes late in the program development, it is necessary to have a temporary method of designating and identifying the individual pages of the program. This is called manuscript pagination.

Experience has shown that it is easier for the program writer to prepare his material and for the editor and other personnel working with the program to read the material if the manuscript is arranged so that the wrong-answer pages follow immediately after the right-answer page to which they relate. This automatically places the next right-answer page at the end of the associated group of pages. This is done irrespective of the arrangement order of the answer alternatives on the right-answer page.

Sub-sequences must follow this same order and must be inserted behind the page from which the first page of the sub-sequence is derived. The pages should be numbered in serial order in the lower right-hand corner. This process is diagrammed in figure 13.



Manuscript Page Numbers 1 2 3 4 5 6 7 8 9 10

Keys RA - Right answer page WA - Wrong answer page

SS - Sub-sequence

Figure 13

Manuscript Pagination

PRELIMINARY EDITORIAL REVIEWS

The Program Edit

Immediately after the chapter or lesson has been written and before it is given an initial student tryout, the material should receive a preliminary edit called the program edit. The program edit is quite limited in its scope and serves a very different purpose from the kind of editing that is done in publishing most other kinds of materials. The program edit may be performed by a technical person (frequently a psychologist), or by the project editor. In either case, it should be concerned only with the following areas.

First, the program editor checks the material against the program outline. Any variations or omissions are noted for discussion with the writer.

Second, the program editor checks the conceptual development of the material against the secondary objectives that have been established for that lesson or unit. This is a qualitative examination in which the editor uses his judgment and experience to determine the material's potential for effectively achieving the stated objectives. In doing this, the editor will form a general impression concerning the adequacy and relevancy of the work. He will determine whether there is a reasonably developed conceptual understanding of the subject matter. He will determine whether the context of the material is useful and appropriate to the subject. Next, the editor will examine the material to determine whether it is in agreement with the assumptions about the learner for whom it is intended. In doing this, he will examine the frames for terms that may require definition. If he finds such a term, he will examine the writer's list of assumed terms and his list of terms that are to be defined. If it is a term that can be safely assumed to be understood, there is of course no problem. If it is not on the assumed list, the editor will examine the "terms to be defined" list to see if it has been defined previously in another lesson or unit. If not, then of course the term should be defined the first time it is used.

Next, the editor will examine the question and answer alternatives in the light of the following questions: Are the examples, the application, and the answer alternatives well chosen and meaningful? Are the answer alternatives really relevant and important to the subject matter? Are all aspects of the subject matter covered by a question? Are the answer alternatives logical and free from ambiguity? Do the questions appear to diagnose and deal with logical errors or mistakes that the student might make?

Some of the questions that the program editor may raise cannot be answered on a rational basis but will have to be settled empirically by testing on one or more typical students. Nevertheless, the program editor should note points on which he has a question so that student data bearing on the problem will not be overlooked. Frequently, such points can be clarified by interrogation of the student after he has completed the program.

Finally, the program editor will check the manuscript to determine whether the planned program strategy has been carried out. Here he will ask such questions as: Has the most appropriate programming technique been employed to achieve each objective? Is the balance between drill and theory development in line with the agreed-upon strategy? Is appropriate remedial material for each answer alternative included?

The purpose of this edit is not to plan extensive revisions in the program, but rather to note any rewriting that may be necessary to make the program effective on a rational basis and to point up areas that will require careful analysis in the student testing.

The Technical Review

In cases where an outside technical consultant is a part of the programming team, the technical edit should be performed before the material is rewritten or student tested. In this way, the recommendations of the subject-matter expert can be incorporated into the initial rewrite.

It should be understood and agreed upon by all parties that the subject-matter expert will limit his comments to his own areas of proficiency. When such an understanding is not reached beforehand, experience has shown that subject experts occasionally become so concerned with the style of the program or technical aspects of the programming process that they fail to perform adequately the function for which they were chosen.

The function of the subject-matter expert should be to determine that the material covers the subject comprehensively within the limits of the objectives and that the material is technically accurate. The person performing the technical edit should give special attention to the answer alternatives and to the content of the material on the wrong-answer pages. Experience has shown that these are two places where the program writer is likely to be weak from a technical standpoint. The subject-matter expert is in the best position to spot illogical wrong-answer alternatives.

In nontechnical programs, the art work is seldom prepared at the time of initial program writing. However, in some technical programs the art work is essential to the development of the material and must be available when the student testing is undertaken. In such cases, the accuracy of the art or illustrations should be checked by the technical editor. This is particularly important in electronics, high-level math courses, and similar programs.

When the technical editor has completed his review of the material, he returns it with his comments to the project editor. It is the project editor's responsibility to make the final decision on which of the recommendations of the technical editor are to be included in the program and the manner in which they will be introduced. The project director has the overall responsibility for the program's effectiveness, and he has a broader view of both the objectives and the technical aspects of the project. He does not ignore the recommendations of the subject-matter expert, but rather translates them into a workable plan for rewriting the individual frames. When the necessary rewriting has been accomplished, the program is then ready for the initial student testing.

The Copy Edit

By the time the program reaches the copy edit stage, there should be no further need for major rewriting or revision of the preliminary version.

If the production schedule of the program is tight, the copy edit can be performed at the same time as the scrambling by using duplicate copies of the manuscript for scrambling purposes. It will, of course, be necessary to transfer the scrambling instructions onto the original manuscript on which the copy edit is performed.

Because the material has now reached the final stages of its preliminary development, the copy edit is limited to the correction of grammatical errors and minor corrections of format. The major tools of the copy editor are his knowledge of the language, of the format of intrinsic programming, and of proof-reading symbols.

Proof-reading conventions found convenient in the preparation of programmed instructional materials are listed in Appendix B. Thorough standardization of this system of communication minimizes the possibility of misunderstanding between the copy editor and the personnel involved in the production of the final manuscript.

SCRAMBLING THE MANUSCRIPT

Scrambling is the process of changing the manuscript of the program into the order or arrangement that will be used in the final presentation. It consists basically of two processes: (1) Rearranging the order of the program pages and (2) Assigning new sequential page numbers.

Scrambling for Book Presentation

The purpose of scrambling material that will be presented in book format is to reduce the student's temptation to move ahead in the program without actually solving the problem or answering the question that is used to test his understanding of the material. The scrambled book does not prevent "cheating" on the part of the student, but it will minimize the natural tendency to "look ahead" on the part of students who are motivated to study and learn the material.

A few simple principles should be reviewed before considering in detail the process of book scrambling. The first page of the program should, of course, be the first page presented to the student. This may or may not be numbered "page one, " depending upon the numbering system used for the title page, table of contents and other prefatory material. The first page should always be an odd-number page so that it will appear on the right-hand side of the open book format. The last page of the program should be assigned the highest page number used in the scrambling process.

It is frequently desirable, in preparing programs for students who are not familiar with intrinsic or scrambled book programs, to provide a reminder on page 2 of the program that the student should follow the page instructions by each answer alternative. This is designed to catch the student who may, from force of habit, complete page one and turn automatically to page two. If such instructions are deemed desirable, then page two should be reserved for this purpose when the scramble is constructed. This will mean that the first lesson or unit of the program will require one more page in the scrambled format than is indicated in the manuscript format.

All of the page numbers between the first and last page assignment of the program should be used. Otherwise, there would be blank pages within the program. As a general rule, the student should not be required to turn more than ten pages forward or ten pages backward to find the answer he has selected. Placing alternatives farther apart than this increases the time lost through the mechanical process of page finding. With these general principles in mind, it is possible to begin the actual process of scrambling.

It would be very inefficient and confusing to shuffle the manuscript physically in developing the scramble sequence, so two very simple forms are used. Figure 14 shows the bubble diagram of a book scramble and figure 15 shows the pagination check sheet. Printed copies of these forms

Project: ____

Scrambler: ______________________________ Date: ______________

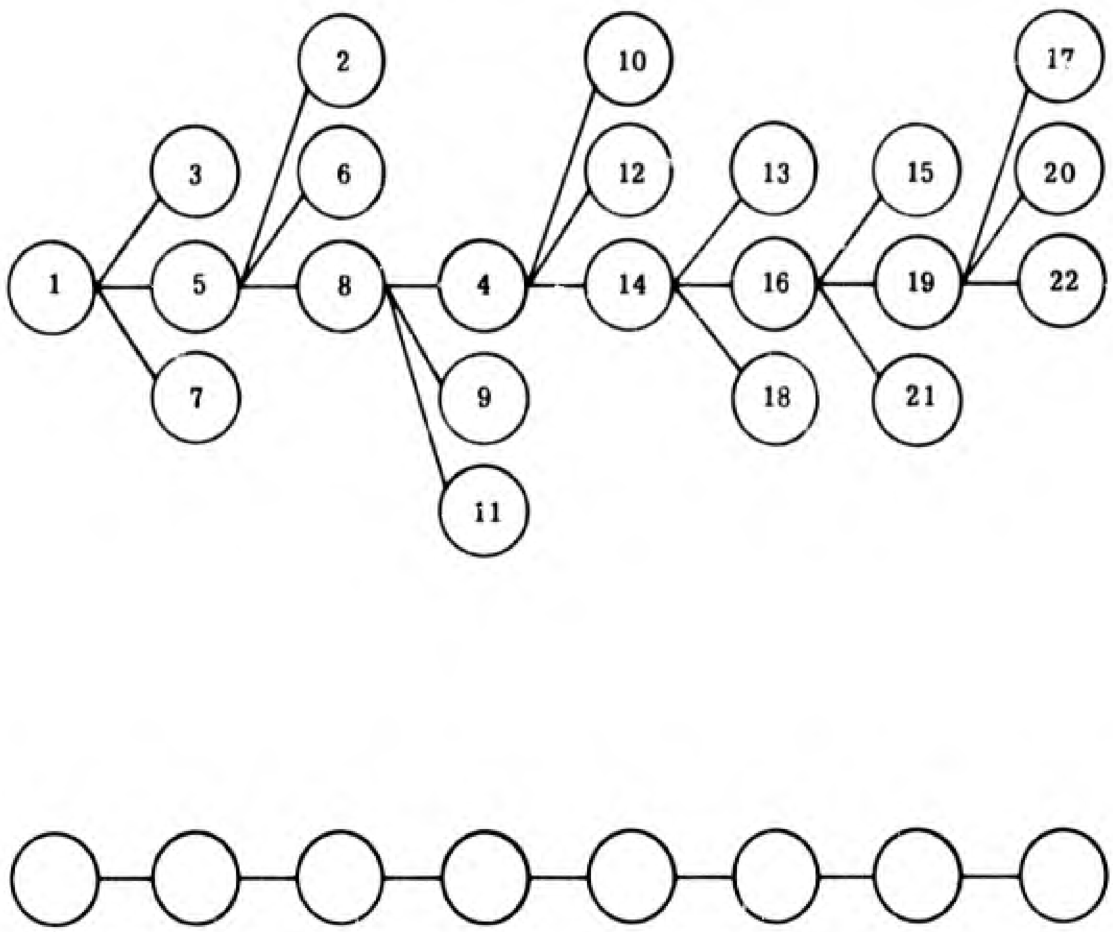


Figure 14

Bubble Diagram of Book Scramble

	~~10~~	~~20~~	30	40	50	60	70	80	90
~~1~~	~~11~~	~~21~~	31	41	51	61	71	81	91
~~2~~	~~12~~	~~22~~	32	42	52	62	72	82	92
~~3~~	~~13~~	23	33	43	53	63	73	83	93
~~4~~	~~14~~	24	34	44	53	64	74	84	94
~~5~~	~~15~~	25	35	45	55	65	75	85	95
~~6~~	~~16~~	26	36	46	56	66	76	86	96
~~7~~	~~17~~	27	37	47	57	67	77	87	97
~~8~~	~~18~~	28	38	48	58	68	78	88	98
~~9~~	~~19~~	29	39	49	59	69	79	89	99

	10	20	30	40	50	60	70	80	90
1	11	21	31	41	51	61	71	81	91
2	12	22	32	42	52	62	72	82	92
3	12	23	33	43	53	63	73	83	93
4	14	24	34	44	54	64	74	84	94
5	15	25	35	45	55	65	75	85	95
6	16	26	36	46	56	66	76	86	96
7	17	27	37	47	57	67	77	87	97
8	18	28	38	48	58	68	78	88	98
9	19	29	39	49	59	69	79	89	99

Figure 15

Pagination Check Sheet

are not essential and blank sheets of paper could serve as well. However, where a large amount of scrambling is done, the printed forms do facilitate the process.

The bubble diagram is essentially a flow chart showing the primary and secondary paths through the program. The printed bubbles on the form are used to record the page numbers of the main right-answer stream. Additional bubbles are added to the chart for the wrong-answer pages and for sub-sequence pages. The pagination check sheet is used to keep a running record of the pages that have been used and the pages that remain available.

The first step in scrambling is to determine the number of pages in the manuscript. The number of scrambled pages will be the same, unless page 2 is reserved for a reminder to follow the special page instructions, in which case one page should be added to the number of manuscript pages to find the number of scrambled pages. Circle the number representing the last scrambled page on the pagination check sheet and scratch through all the numbers higher than this one. The remaining numbers are available to work with. Next, put the number 1 in the first bubble of the bubble diagram, put number 1 in the upper right-hand corner of the first page of the manuscript, and put a single slash mark through the number 1 on the pagination check sheet.

Find the right answer alternative for the first question. For the purpose of illustration, assume that the correct answer is the first of three alternatives. This answer page could be assigned any number from two through eleven, but for the purpose of this discussion, suppose it is assigned the number five. The figure 5 would then be written in the second bubble of the bubble diagram, after this alternative on the first page of the manuscript, and in the upper right-hand corner of the manuscript right-answer page. Finally, a single slash would be drawn through the figure 5 on the pagination check sheet.

The wrong answers can now be handled. As they happen to be the second and third alternatives in this example, they will appear on pages with higher numbers than the right answer, which was the first alternative. Page numbers are always allocated in sequence so that the student gains no hint of the answer from this source. Suppose the wrong answers are allotted pages 8 and 10. Two bubbles are then constructed below the bubble representing page 5 on the bubble diagram, the figures 8 and 10 are inserted in them, and single lines are drawn back from each of these new bubbles to the bubble representing the question page, page 1. The numbers 8 and 10 are recorded beside the wrong-answer alternatives on page 1, and in the upper right-hand corner of the manuscript wrong-answer pages, and the numbers 8 and 10 are scratched out on the pagination check sheet.

The next question appears on page 5, the right-answer page for the previous question. It should be kept in mind that right-answer pages are usually also question pages. Suppose the right answer for this question is the second of three alternatives, and we assign it to page 7. The figure 7 is written in the third bubble of the bubble diagram, after the right-answer alternative on page 5 of the manuscript, and in the upper right-hand corner of the manuscript right-answer page. The figure 7 is scratched out on the pagination check sheet. Now we are ready to handle the wrong answers for the second question. The page assignment for the first wrong-answer alternative should be a lower number than the right-answer alternative which was assigned to page 7. Pages 3, 4, and 6 are available (2 was used for the reminder about following a scrambled text), but 4 cannot be used because it is facing the question page, and 6 cannot be used because it is facing the right-answer page. Page 3 will have to be used for the first wrong answer. A bubble is constructed for figure 3 above the bubble representing page 7 (the right-answer page), a single line is drawn back to the question page (5), and the figure 3 is recorded beside the first alternative on the manuscript question page and in the upper right-hand corner of the manuscript wrong-answer page. Finally, the figure 3 is scratched out on the pagination check sheet.

The second wrong-answer alternative, which was listed below the right-answer alternative on the question page (5) will be given a higher page number assignment. The other steps will be as previously outlined.

This same process is repeated until all the pages of a manuscript have been accounted for. In instances where a sub-sequence is used, the first frame of the sub-sequence is diagrammed as a wrong-answer frame but on the connecting line an arrowhead pointing toward the sub-sequence frame is indicated. Then, the first frame of the sub-sequence is treated as a right-answer frame with other right- and wrong-answer frames branching from it. The line indicating the student's path from the last frame of the sub-sequence is drawn to show the return into the mainstream of the program.

After the bubble diagram has been completed, it should be checked against the pagination check sheet. This is done by making a second check on the pagination check sheet for each number represented in the bubble diagram. When this check is completed, there should be two check marks for each page number on the pagination check sheet. At this point the book scramble is complete.

Scrambling for Presentation on the AutoTutor Mark II

Scrambling for the AutoTutor Mark II consists of arranging the manuscript for filming so that the correct sequence of images appears as the student pushes the buttons that indicate his selection of answer choices.

The first step in the process is to construct an unscrambled bubble diagram. This is merely a flow chart of the right- and wrong-answer pages as they have been prepared in manuscript form. It can be done on the same bubble diagram form used for book scrambling. The difference is that the bubbles will represent manuscript page numbers rather than scramble page numbers. A simple thirteen-page bubble diagram is shown in figure 16. Notice that although the right-answer choice may be the second of the three given, its page number is always the highest number of the three.

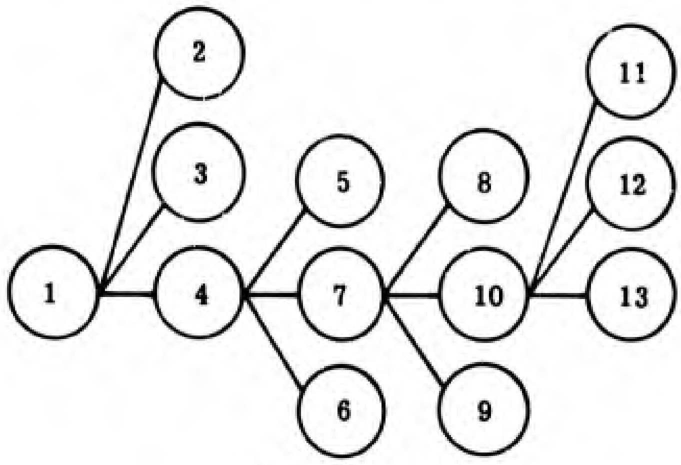


Figure 16

Sample Bubble Diagram for AutoTutor

There are two phases to scrambling for the AutoTutor Mark II. One is the arrangement of the copy in the proper sequential order. The other is coding each frame to control the student properly.

This student control is accomplished by providing the lower border of each frame of film with a pattern of lightness or darkness that operates or blocks photoelectric cells located in the AutoTutor. This coding determines which buttons the student will be able to operate when a given frame is displayed.

If the film is clear, only the "I" button will operate. This condition is coded W.

If the film border is all dark, all of the buttons will operate. This pattern is called code X.

If the right third of the film border is dark, only the RETURN button will operate. This condition is coded Z.

If the left two-thirds of the border is dark, the RETURN button will not operate. This condition is coded Y.

If special control of the student is not needed, the frame is coded X, which means that all the buttons will operate. The student may return to review the previous image without penalty, for the error counter does not record when the frame is coded X. He may move forward to any of frames A to H or he may use the I button to move backward in the program 19 frames. Generally speaking, the X code is used for "right-answer" images. About a third of all frames are coded X in the average program.

When the directions on a frame read, "Now return and make another choice," the frame should be coded Z. This coding is used for wrong-answer pages that do not begin sub-sequences. When the student has erred, he is told how he has erred and is provided with a further explanation. He is then sent back to the question to try again. In such a situation, the student should not go ahead to a new idea, nor should he be permitted to wander around in the program looking for the right-answer frame. His activity is controlled by the use of the Z code which disconnects all of the buttons except the RETURN button. About 50% of all the frames are coded Z.

The Y code prevents the RETURN button from operating. This code is ordinarily used on the first frame of a sub-sequence. In this situation, the student is to move forward into the sub-sequence instead of returning to the previous frame and selecting another alternative. Coding the frame Y prevents the return button from operating and forces the student forward through the sub-sequence.

The W code permits only the I button to operate. The I button reverses the film and moves it back 19 frames at a time. The primary use of the I button is to permit review of whole sections or sequences of the program. The I button is also used for other special purposes that will be described at the appropriate point in the scrambling process.

The coding on a frame is hidden from the student and he has no way of knowing how much freedom is available. In most cases, this is not important because the student is given EXPLICIT instructions on each frame.

The physical characteristics of AutoTutor Mark II are such that if a student is on an odd-numbered frame, he can move only to an even numbered frame and vice versa. The A button advances the student one frame; the B button advances the student three frames; the C button advances the student five frames; the D button advances the student seven frames, etc. Button H advances the student 15 frames and button I moves the student 19 frames in reverse.

The RETURN button operates a memory circuit in the AutoTutor in such a fashion that the machine retraces its most recent excursion and displays the last previous image viewed. The RETURN button will operate only one time. If the student presses the RETURN button a second time in succession, there will be no further action.

The task of the scrambler is to arrange the branched program into a sequence of linear frames. The tools of a scrambler are several colored pencils, a scramble rule, and a large sheet of quarter-inch graph paper or special scrambling paper.

Figure 17 represents a simple bubble scramble with the button letters indicated for each page. The button directions on page 1, which will also be frame 1, are "choose A, B, or C." Button A would advance the film one frame which would put manuscript page 3 on frame 2. Button B would advance the film three frames which would put manuscript page 4 on frame 4. In like manner, manuscript page 2 would become frame 6.

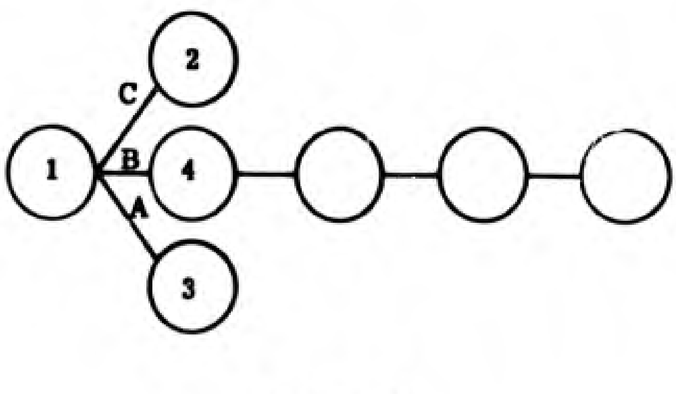


Figure 17

Sample Bubble Scramble

To keep track of this process, which quickly becomes very complex, a sheet of either regular graph paper or scrambling paper is useful (see figure 18). A complete sheet has 400 numbered squares on it and four such sheets are required for scrambling a full 1500-frame reel of TutorFilm.

It is the practice of the scramblers at the Educational Science Division to indicate the location of a question frame with an X under the frame number. The location of the answer frames is indicated by the letter designation required to reach them and noted under the frame number. Thus, our example shown in figure 17 would be like figure 18 when transferred to graph paper. Notice that the button letters are placed on the same horizontal line as the X that indicates the question. (It should be kept in mind that right-answer frames are usually also question frames; they carry the mainstream of the instructional sequence, as the bubble diagram shows.)

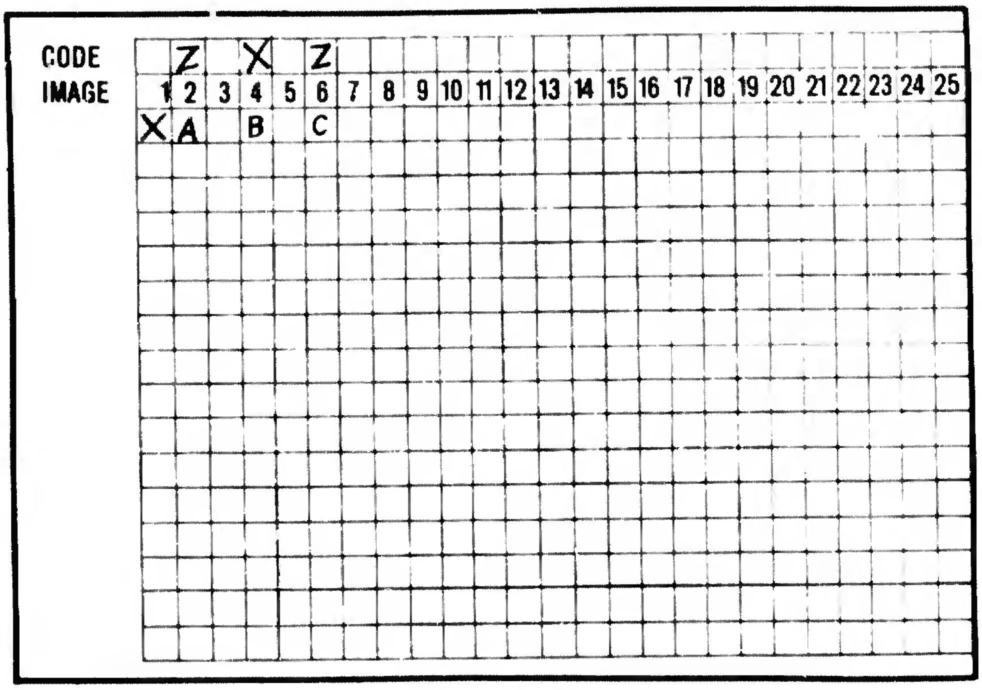


Figure 18

Portion of Scramble Sheet

On the bubble diagram (figure 16), page 4 (second alternative) was shown to be the correct answer alternative. On the scramble sheet, this became image number 4 (reached by pressing button B). Since it is a right-answer frame, it should be coded X. This X coding is shown on the scramble sheet above the frame number.

Frames 2 and 6 are wrong answers and should be coded Z. The letter Z appears above the frame numbers 2 and 6 in figure 18.

Figure 19 shows a completed bubble diagram for a thirty-four page lesson. Figure 20 shows the completed scramble diagram for this same lesson. From this point on, these examples will be used to illustrate the scrambling process. Frame 1 of figure 20 is the same as page 1 of figure 19. Remember that an X is placed on the scramble sheet to indicate a question frame; therefore, an X has been placed under image number 1 in figure 20.

Also, note in figure 20 that image 1 is coded Y. This is one of the special uses of this code. Since this is the first frame of the program, there is no prior frame to which the student might return. Also, the student is not to press the I button, because it will move him backward into the leader of the film. He is to move forward only. The Y code is used for this purpose.

The next question frame is image number 4. This indicates that image 4 is the right-answer alternative from image 1. Reference to the bubble diagram confirms this because the second answer alternative from page 1 is the correct answer. As this is the correct answer, it is coded X.

In the discussion of scrambling for book presentation it was stated that the manuscript page

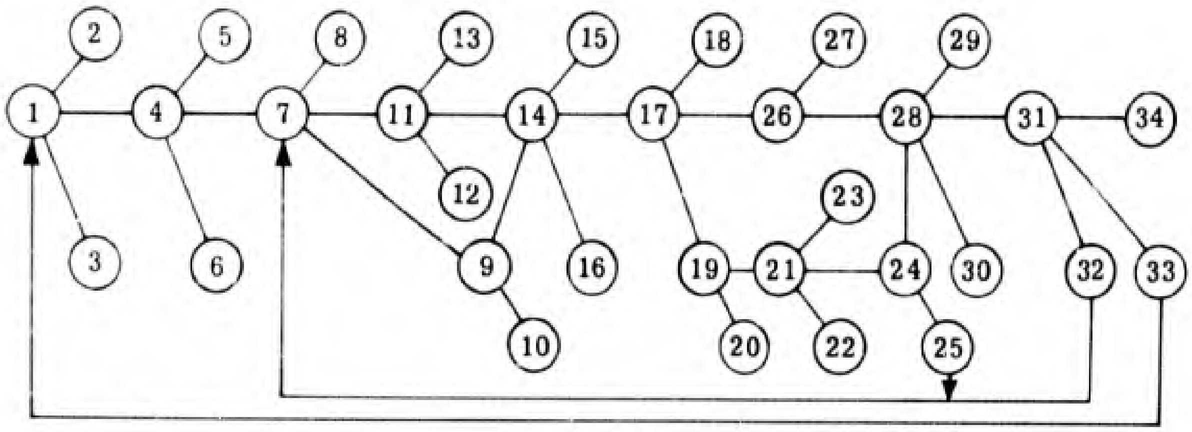


Figure 19
Completed Bubble Diagram

number was placed in the lower right-hand corner and the scramble page number in the upper right-hand corner. In scrambling for the AutoTutor Mark II, the same procedure is followed except that in this instance it is the image or frame number that is recorded in the upper right-hand corner. Sometimes the same manuscript is scrambled for both book and machine. In this case, the frame number for the machine scramble is indicated with a colored pencil.

The code directions should also be indicated on the manuscript in the lower left-hand corner by the letters W, X, Y or Z.

On the bubble diagram (figure 19), note an arrow drawn from page 33 back to page 1. This means that the student must be able to return from page 33 to page 1. To accomplish this, use is made of the I button's capability of moving the film 19 frames in reverse, and what is known as an "I track" is constructed. An I track consists of a series of frames connected by virtue of location. Each is 19 frames away from the next. Such frames are usually coded W, the coding that permits only the I button to operate. Thus, when a student arrives at a point on the I track, he has no choice but to move to the prescribed destination, which is some multiple of 19 frames in reverse.

In this example it is necessary to return the student from page 33 to page 1. For this to be done, every nineteenth frame must be reserved for this I track. Later, another I track will be needed to return the student from pages 25 and 32 to page 7, which on the scramble diagram also happens to be frame 7.

The first frame that should be reserved for the I track which will return to frame 1 is frame 20. If the student is at frame 20 and presses button I, the AutoTutor will move the film to frame 1. The next reserved frame is 39. This is the frame or image on which page 33 of the bubble diagram must be placed. It is the usual practice at the Educational Science Division to shade these I-track frames with a colored pencil.

From page 4 of the bubble diagram, there are three alternatives of which page 7 is the correct answer. On the scramble diagram, these alternatives become image 5, 7, and 9. Image 7 is the right answer, and as it is a question page, X is placed below the letter B.

A means must now be provided for moving a student from page 32 on the bubble diagram back to page 7. Thus, the next step in this scramble is to reserve another I track. In constructing this second I track, the first frame to be reserved is frame 26 (7 + 19 = 26). Since the page that will lead back to image 7 is 32 on the bubble diagram, an additional jump in the I track is needed. This means that the next image for this second I track will be number 45. Forty-five is then the frame number that will be assigned to page 32 of the bubble diagram.

Since it is very easy to make an error in counting frames in the process of scrambling, the Educational Science Division has developed a useful tool called the scramble rule. The scramble rule (figure 21) is constructed by mounting a piece of scramble paper on a straight edge, marking

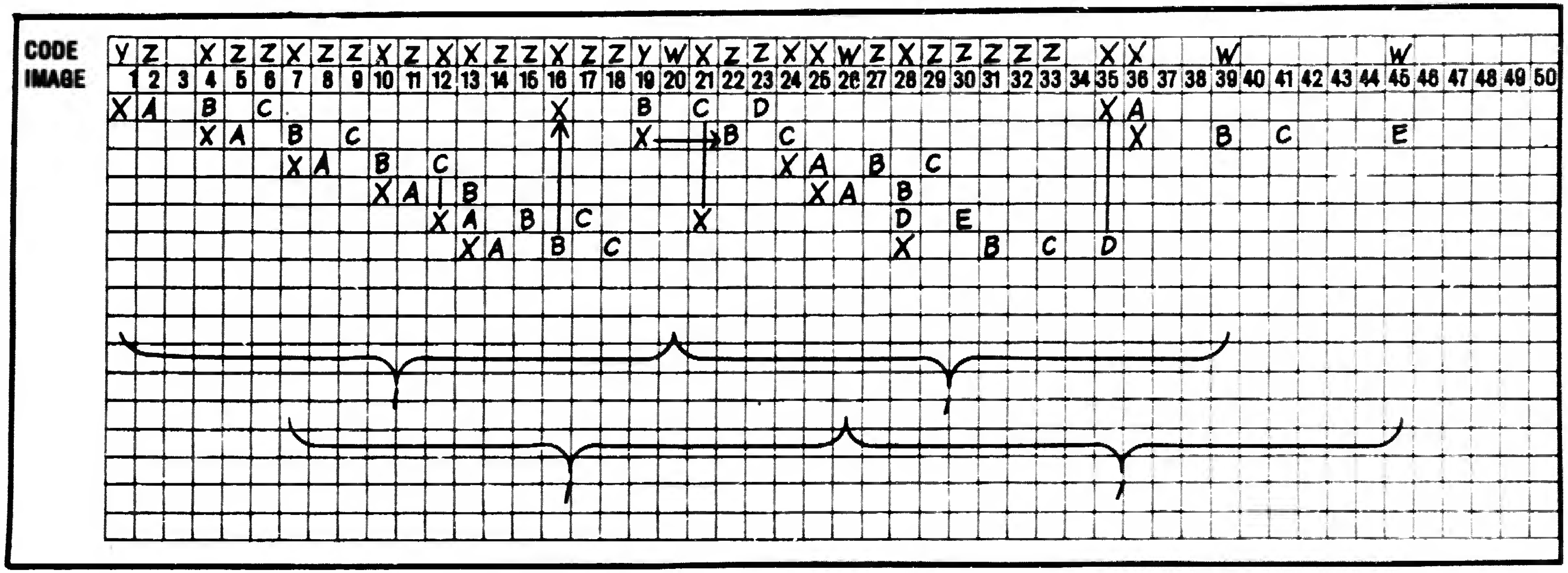


Figure 20
Completed Scramble Diagram

a home point with an arrow, indicating with the proper letter the relationship of frames accessible to home.

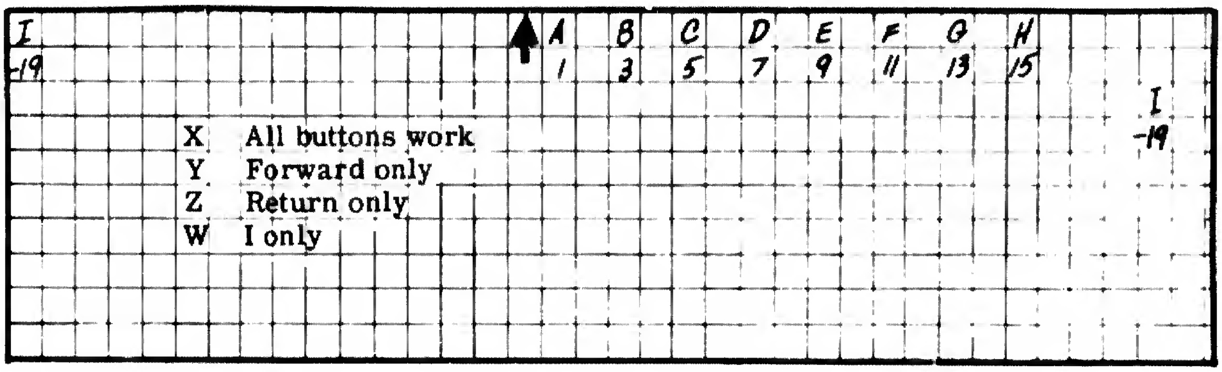


Figure 21

Scramble Rule

To use the scramble rule, place the arrow under the frame being worked on, and read directly the location of the frame reached by the use of each of the buttons, and also the location of tne next succeeding I track. From frame 7, there are three alternatives which have been assigned buttons A, B, and C. These represent pages 8, 11, and 9 on the bubble diagram. It will be noted that both pages 11 and 9 are question pages. Therefore, both buttons B and C will have an X placed beneath them. However, it would be confusing to have more than one question frame located in close proximity on the same line. Therefore, the X for page 9, which is shown as frame 12, is dropped down one line on the scramble sheet below the X for page 11 which becomes frame 10.

The next critical problem in this example is represented by page 14 of the bubble diagram. The students who arrive at both pages 11 and 9 must be able to press a button that will bring them to page 14. Therefore, page 14 must be placed far enough away from pages 9 and 11 so that there will be room for both tracks to reach it. This was done in this example by placing page 14 on frame 13 of the scramble diagram. Thus, page 14 (frame 13) is reached from page 11 (frame 10) by pressing button B, and from page 9 (frame 12) by pressing button A.

A question may arise at this point concerning what the machine would do if the student after reaching page 14 (frame 13) were to press the R button to review the previous frame. This is not a problem because the memory circuit in the machine always carries the student back to his last excursion. Thus, the student who arrived on page 14 from page 11 would return to page 11 when the R button was pressed. In a similar manner, the student who arrived on page 14 from page 9 would be returned to page 9.

The next problem that must be attended to is an extended sub-sequence. Note on the bubble diagram that there are two paths to reach page 28 and that page 24 needs to be connected through page 25 to the I track that goes back to page 7. To begin with, page 25 will be placed on the closest I-track frame to page 7. This will be frame 26, which will then be coded W. The next step is to move backward to locate page 24. We will use the A button for this purpose, to place page 24 on frame 25. Page 28, the other alternative from page 24, will then be assigned to the B button and will become frame 28.

After this critical part of the program has been set up, it is possible to return to the sequential development of the scramble. As these preassigned frames are approached, it will be necessary to make certain that they are led into with the proper button instructions. Thus far, this discussion has focused on the mainstream and the critical sub-sequence and wrong-answer pages. Naturally, the other wrong-answer pages will be assigned to frames at the same time that these critical points are located.

Page 17 of the bubble diagram comes from page 14, which has been previously located as frame 13 on the scramble diagram. Since page 17 is the second alternative on page 14, it will be assigned the B button. This places page 17 on frame 16 of the scramble diagram. (Because the scramble is extending rather far down on the scramble paper at this point, a connecting bar can be drawn up to the top of the page in the column of frame 16 and an X can be placed there to indicate the location of the question at the top of the column. This procedure permits a fresh start at working down the paper.) Page 18 now becomes frame 23 and page 19 becomes frame 19. The right-answer alternative, page 26, becomes frame 21. Page 19, which is frame 19, is the entrance into a mandatory sub-sequence. Therefore, it should be coded Y so that the student will be forced to move forward only.

Next there is the problem of locating page 21, the right answer from page 19, so that it can be reached from frame 19 and will lead to page 25, which has been previously located. This is accomplished by assigning the C button to page 21, to make it frame 24. Then the A button will become the assignment for page 24 (the right-answer page from page 21), which has been previously located at frame 25. Wrong-answer page 22 will be assigned the B button, placing it on frame 27, and wrong-answer page 23 will be assigned the C button, placing it on frame 29. The scramble rule greatly simplifies the location of these points.

Page 28 (frame 28) has already been located. Page 32 has previously been located as frame 45 and page 33 has been located as frame 39. These are the beginnings of the I track. The next problem is to locate page 31, which is the right-answer alternative to page 28 and the lead-in to pages 32 and 33, at a point on the scramble diagram so that it (page 31) can be reached from page 28 and will lead to pages 32 and 33. The scramble rule shows that this is not possible with a single move. The solution to this problem is to add a "jumper frame" to tie the two requirements together. This is accomplished by assigning the letter D to the right-answer alternative for page 28 (frame 28). When the student arrives at this frame (frame 35), he will be instructed to "Press button A." This will place him on frame 36, where the information and questions from page 31 will be provided.

The wrong-answer alternatives from these pages provide no problem in scrambling. They are merely assigned button numbers above or below (depending on their placement on the page), the button or frame assignments of the critical right-answer pages.

To complete the scramble, copy must now be prepared for the frames that were not filled with pages prepared by the writers. The first of these is frame 3. There is no legitimate reason for the reader to arrive at frame 3, so a page is prepared that reads, "You have arrived here by mistake, please press the RETURN button," and the frame is coded Z.

The next vacant frame is 20. This frame is an I-track frame coming from frame 39. On this frame will appear the instructions, "Please press button I one more time," and the frame will be coded W so that only the I button will operate. Frame 26, which is part of the other I track, is a point that can be reached from frame 25 or frame 45. In this instance, the W code will still be used but the direction should read simply, "Press button I." Frames 34, 37, 38, 40, 42, 43, and 44 are "return" frames and will contain the directions, "You have arrived here by mistake, please press the return button."

On frames 39 and 45, the student is locked into the I track. Since he has no alternative but to press the I button, he may be told that he is wrong and given the reason for his being sent back. Then, of course, he must be told to press button I.

Frame 46 is the usual location for the first page of the next chapter or lesson. In this scramble, it could have been frames 42 or 43. However, if frames 42 or 43 were used to begin the next lesson, the two lessons would be inter-leaved. This might cause difficulty if revisions in one of the lessons were required.

This is not the only way that this particular lesson could have been scrambled. However, this lesson does present a good example of the typical problems that face the scrambler and practical solutions for each of them. Scrambling is an exercise in logic and skill in the technique which comes only with practice. Many of the obscure points that have been covered in this demonstration become meaningful when the actual process of scrambling is performed.

PRE-PRODUCTION EVALUATION

Preliminary Student Testing

Between the technical edit and the rewrite edit, the material is given a preliminary student testing. The first informal evaluation is usually conducted under the direct supervision of the program writer. At this point, the primary interest is in the qualitative material that can be gathered from the student rather than in objective performance results. The informal evaluation provides information that will assist the program writer in discovering items that are confusing, difficult, repetitive, or particularly helpful. This evaluation gives the program writer some indication of the time needed to complete the sequence and of the student's reaction to the program. Also, this information frequently provides initial guideposts for more extensive field validation.

Analysis of the responses and errors is one part of this evaluation. The Program Record Form, shown as figure 3 on page 20, is useful for accumulating this information. The student is asked to record his progress through the program by writing the page numbers in the blanks as he selects them. If the student is returned to a previous page, he records its number again each time he reviews it. Provision is also made for recording the student's comments. These reactions will consist of phrases such as "I didn't understand the question," "The wrong answers are obvious," "You didn't give us the information needed to answer the question," "I didn't notice the word not in the question," etc.

This type of information, coupled with the interview which the program writer will conduct with the students and his observation of the students' behavior as they work on the program, will help to clarify many points in the program writer's mind. If his assumptions concerning the students are faulty, that fact will become obvious if the students have been properly selected. If he has not correctly translated objectives into the kinds of behavior needed, that will become apparent. If the method of programming does not suit the subject or the student, that fact will also become apparent.

If the students make few or no errors in this preliminary trial, it should be a matter of concern to the programmer. It probably indicates that his assumptions concerning the student were faulty or that the answer alternatives were too easy and failed to fulfill their diagnostic requirements. If the error rate is very low, but the program objectives are achieved, the fault is probably in the assumptions concerning the students; or at least it may be concluded that the questions and the answer alternatives are not fulfilling their function.

Often this testing is conducted with the original manuscript. However, if 6 or more students are used, it is frequently desirable at this point to reproduce several copies of the program so that comments and changes can be made on the manuscript during the interviews with the student. Whether this preliminary testing is done in-house or on-site will be determined largely by the availability of students. It is more economical of the programmer's time to have the students come in at scheduled times. However, if the writing is not being done on-site and typical students are not obtainable locally, the programmer may have to go to the students.

The results of this testing should be made known to the editor, who will note the information concerning vocabulary and style, the reading level and tone of the material. Subjective reactions of the students are carefully evaluated. The editor will want to know not only the students' general reaction to the unit but also many specific reactions. Were the questions challenging? Which questions seemed too difficult? Why? What alternatives seemed obviously right or wrong? What mental processes did the students use in selecting alternatives? Did the student really work the problems, or did he select alternatives by a process of elimination? Did the student feel the program writer was talking down to him or talking above his level? Did the program have variety and sparkle or did it seem dull and academic? Were the review items a test of rote memory or did they challenge and stimulate thought? Was there sufficient graphic material and did the illustrations contribute to an understanding of the concept? Were there terms or questions that seemed confusing or ambiguous? How successfully was the student able to achieve the secondary objective for that unit after completing the program? The information provided by these kinds of questions, along with the objective data gathered from the student testing, should be kept clearly in mind by the person performing the rewrite edit.

The Rewrite Edit

Using the information gained from the preliminary student testing, the editor rephrases and polishes the text material on frames that do not require a major rewrite, and suggests improvements on frames and sections that are to be rewritten. Special attention is paid to the adequacy of explanations provided on the wrong-answer pages.

Special attention also is paid to sub-sequences and particularly to the frames that lead into them. At this point, sufficient data has been accumulated to determine on an impartial basis whether or not a sub-sequence is needed, but attention should now be given to the alternatives leading into the sub-sequences. If the sub-sequence was used extensively in the preliminary student testing, the editor should attempt to determine whether the answer alternatives leading into it are tricky or ambiguous. If no use was made of the sub-sequence, but it appears to be needed on a rational basis, then the alternatives leading into the sub-sequence should be examined to determine whether they are in fact diagnostic of the needs to be filled by the sub-sequence.

The Writer's Final Edit and the Proofing Tape

After the rewrite edit, there are two possible ways of handling the manuscript. One way is the traditional approach, in which the manuscript moves from the copy editor to the Production Department, where the final typing and art work is completed. The manuscript is then read with the animation copy by two proof-readers working together, the errors are corrected, and the final production copy is returned to the program writer for final approval. The usual result with this method is that the program writer, having the final responsibility and wishing to turn out a perfect product, does some rewriting and makes it necessary for certain of the editing and production processes to be repeated. Another disadvantage of this method is that the two proof-readers must work as a team, and if one is interrupted the other loses valuable time.

In an improved method of handling the final stages of the check-out, the manuscript moves not from the copy editor to production, but rather from the copy editor back to the program writer. Using a tape recorder or dictating machine, the program writer reads the manuscript aloud in the same manner as the first member of the proof-reading team; that is, he reads all the punctuation, editorial corrections, and special production instructions.

This critical reading on the part of the program writer frequently uncovers errors that would go undetected until after the final production was complete. When such errors are discovered, the program writer merely stops the tape, makes the correction on the manuscript, and reads the correction into the tape. The result is that when the tape is complete, the program writer is satisfied with the material as it stands. If the program writer finds editorial changes which have modified his meaning or with which he disagrees, these can be straightened out before the program reaches the production stage.

When the program writer is satisfied with the material and has completed the preparation of the proof-reading tape, the manuscript is then sent to production. The proofing tape is set aside for later use.

When production is complete, a single proof-reader then takes the tape and a tape recorder and the finished production copy. If a discrepancy appears between the tape and the production copy, he refers to that page of the original manuscript which he also has at hand. Overlay paper is used for corrections that must be made on the production copy.

The most apparent advantage of this system is that it requires only one proof-reader. If he is interrupted, he simple stops the tape recorder and the time of a second person is not lost. A second and even more important advantage of this system is that many changes which the program writer would not normally see until the program was in final form or which might be totally missed by a rapid silent reading of the finished copy are detected in the preparation of the proofing tape and are corrected on the manuscript before the final work is done.

This system does require the time of the program writer, who is usually a higher-salaried person, to replace the equivalent amount of time of the second proof-reader. At first glance, this would appear to add to the expense of this process. However, the time used by the program writer in preparing the proofing tape is usually only about half again the amount of time usually required for him to give the program a silent reading and (supposedly) careful consideration. Thus, if the proofing and checking of a lesson by the traditional system required 10 hours each from two proof-readers and 8 hours from the program writer, the total time for this process would be 28 hours. Using the improved method, the same lesson would require 10 hours from one proof-reader and 14 hours of the program writer's time, for a total of 24 hours. The salary difference between the second proof-reader and the program writer would be offset by the savings in errors detected and corrected before the production process was undertaken. Also, the program writers who have used this system find, with experience, that they are more satisfied with their work and are more willing to accept the responsibility for the final product than under the old system.

PRODUCTION OF FINAL COPY

When the writer's final edit and the preparation of the proofing tape are finished, work can be begun on the production of the final copy for filming. Preparation of the production copy is followed by scramble checking, filming, and film checking.

Preparing Production Copy

All TutorFilm material should be typed on animation paper. This is a paper of very good quality which is specially printed with registration holes, code bars, and guide lines to indicate the maximum usable typing area. The type style used should be one that will reproduce well through the microfilming process; that is, it should be at least a 12-point type, without any hair-lines. It is desirable to use special type characters rather than hand-made symbols wherever possible.

Manuscript copy delivered to the production department should be scanned by the art section before final typing and the amount of space required for art indicated in the margin. After typing, button decals and simple line work are put directly onto the animation pages. Button decals, used in the text and following answer choices, are applied from "burnish down" lettering sheets. When a complex illustration is repeated on several pages, photostat copies can be made from the original art work and spliced onto the animation page.

The animation copy is proofread after production is completed[1], and the pages requiring corrections are returned for correction. If there are only a few lines to be changed, the correction may be done by splicing in the retyped lines. This procedure saves retyping and proofreading time.

Scramble Checking

The final stage in the production of a course is the "scramble check," which is a check on the scrambling of the course and, to a degree, on its content. Since this is the final step, there are several important details which must be checked.

Before the scramble check is begun, the animation copy must be arranged in order by frame number. If the copy has been properly scrambled and produced, there will be one frame for each number in sequence, and no number will be used more than once.

The first item to be checked on each page is the code. Page 1, ordinarily the "How to Operate the AutoTutor" page, must be on a Y code, so that the student cannot back the program into the leader of the film by pressing the R button. Now the typed copy must be checked to see that it is within the blue guide line on the right-hand side of the page, and that it does not exceed the margin guides at the left, top, or bottom of the page. These margins can be checked on every page by a quick glance.

---

[1] See Appendix B for proofreading symbols.

Next, the button instruction at the end of the copy on the page must be checked to make sure that there is a button decal or a typed button instruction there. The button instruction on page 1 is probably "Press Button A." Page 2 should be inspected next, since the A button will advance the film one frame, and the margins of the copy, the coding of the film, and the next button instruction should be checked. The next page to be checked will be the one indicated by the button instruction on page 2.

Following the front matter of the course is the body of material which comprises the programmed course itself. The pages of this copy will present material followed by a question that will have several different answer choices.

For example:

> 747
>
> Read the following sentence:
>
> The ancient masks represented various abstract forces—death, love, fertility, and honor.
>
> What punctuation mark does the dash replace in the sentence above?
>
> | | |
> |---|---|
> | A semicolon. | C |
> | A colon. | B |
> | A comma. | A |

The above question was asked on page 747. The code on the page is X. This code will work because all buttons will operate on an X code. Next, each of the answer pages must be checked. On page 748 (choice A from 747), the answer should read:

The dash replaces a comma in the following sentence.

The ancient masks.....

The example sentence must appear with every word and mark of punctuation on the answer page exactly as it appeared on the question page. If choice A is not the correct answer, the button instruction at the bottom of the page will usually be "Press Button R" or "Please RETURN." For such instructions, a Z code is called for, since only the R button will operate on a Z code. After comparing the answers, checking the code, and checking the margins, it is advisable to place a small blue mark in the upper right-hand corner of the animation page to indicate that the page has been accounted for.

Similarly, pages 750 (choice B) and 752 (choice C) must be checked. One of these pages will be the right-answer choice which contains another question and set of answer choices. These will be the next frames checked. (As the copy is being gone through in this manner, an answer choice may appear to be misplaced. This is probably an answer choice reached by pressing the I button from 19 frames ahead.)

In some cases a wrong answer to a question will send the student back to review previous material. The button instructions must send him back to the appropriate place. To go back, the student will ordinarily have to push the I button one or more times; the I button rewinds the film 19 frames at a time, and in order to continue an I-track, an I button page must be placed on the animation copy every 19 frames. For example, to return a student to page 7 from page 83, there must be a page reading "Press Button I" on frames 83, 64, 45, and 26. These frames must be coded "W."

All of the above steps become automatic, and can be accomplished by one person in a short time. It is often advisable for one person to handle the above steps while another person makes the record of the scramble check, although one person can do both jobs without too much difficulty.

To make this check of the scramble, the regular scramble pads are used. When completed, this sheet is filed as a record of the check.

A scramble check is not so thorough as the original scramble with its elaborate color-coding. The person checking the animation copy will call out the code of the page to his partner (assuming that two people are doing the check), who will write it above the frame number and then mark the button instruction on the appropriate frame column. In the example on the next page, frame 747 is an X code, with button instructions A, B, and C. The button which is on frame 747 indicates choice C from frame 742. (The X underneath indicates that a new question is presented.) In the case of wrong-answer frames which are on Z codes, a Z is marked above the frame number, as in figure 22.

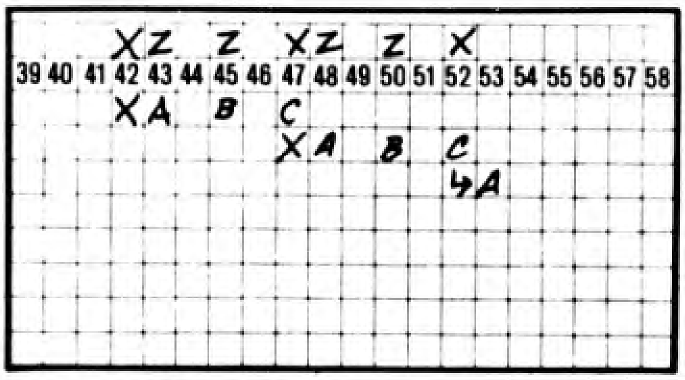


Figure 22

Beginning of Scramble Check

Pages which are not part of the textual matter of the course are "mistake-RETURN" pages and are indicated with a Z above the frame number. These frames are shown as blank on the diagram above. (If two people are doing the scramble check, the person checking the animation copy should say "mistake" when a mistake-RETURN page is encountered. Otherwise, he says "Z" for a wrong-answer page. In both cases a Z is written above the frame number.)

When a code letter and its corresponding button instructions are given for an empty column which contains no button letter, it is necessary to count forward 19 frames to see whether it came from an I button. If it is an I button answer choice or instruction, the problem is solved. However, if there is no I button, then an error in scrambling has been made. If the person who prepared the original scramble is not available for consultation, the scramble checker must read the page carefully and look for its source page in order to correct the error.

Frames which contain the I button instruction on an I track are marked with a W at the top of the column and an I in the first empty square below it; then a wavy red line is drawn down the length of the column as in figure 23. Another I-track frame should be found and marked off 19 frames before the first. Draw a red line between 4 and 23 with arrows going both ways at its ends.

Any problems arising as a result of the checking should be resolved as they are discovered.

At the completion of the scramble check, the frame order should be checked once more to make sure that all pages are still in order for filming.

Filming Requirements

TutorFilm negatives must be made by a vendor capable of microfilming with a 35mm sprocketed camera.

TutorFilm copy must be typed on animation paper with pre-printed code bars and with the 3 holes punched on the left-hand side. These holes fit over a peg bar on the table of the microfilming stand. Each page is placed on the peg bar, one after the other, and exposed. The use of the peg bar insures the registration of each frame to each other and to the film. Most films are shot at a ratio of 13:1, with the density of the negative between 1.45 and 1.65. Film used is standard 35mm, double sprocketed. Negative quality must be excellent.

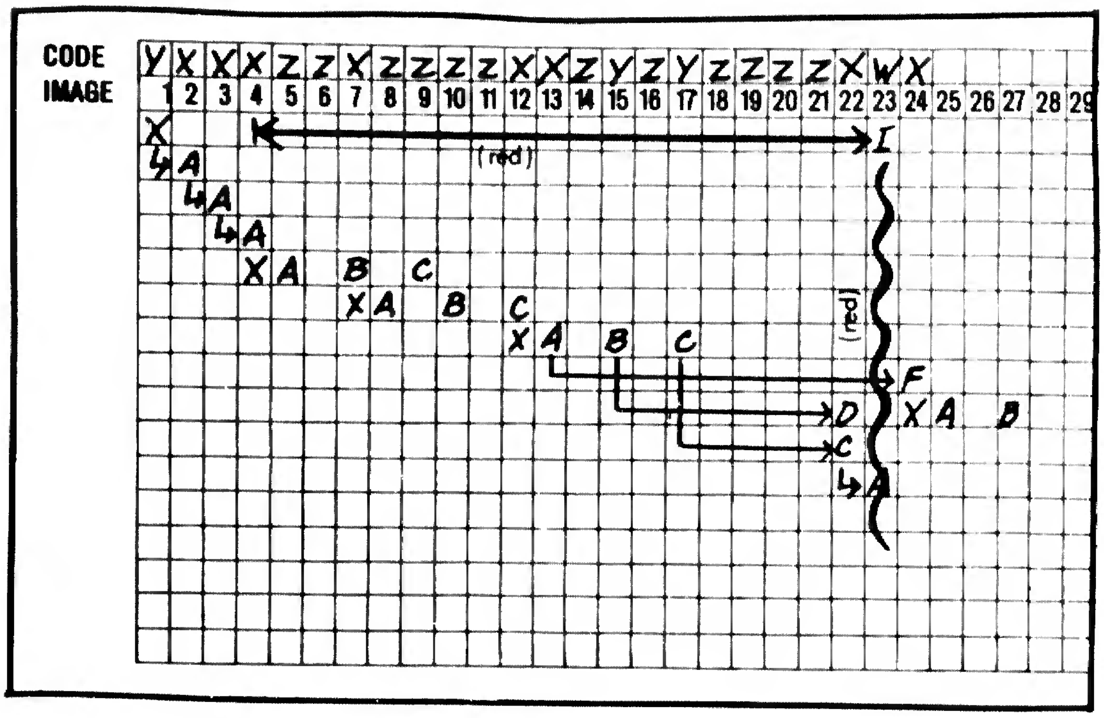


Figure 23

Scramble Figure with I-Track

Prints must be made using a sprocketed printer to keep frames in register with the negative. Prints should be on 100' reels, with square spindle holes on both sides. Prints should be wound with the emulsion side out.

Film Checking

Before negatives are released for distribution prints, test prints must be checked and inspected. The film laboratory supplies a negative and three test prints (1 light, 1 medium, 1 dark). Normally the medium print is best, but at times the light or dark print may be better.

The film checking procedure is as follows. The first check is for proper sequence. Starting with the normal print, check only the page numbers to make sure that they are all there and that they are consecutive. To perform this check, keep the Code Release button depressed and use only the A button, which advances one frame at a time. Sequence checking is important because any frame shot twice or left out by the microfilmer will upset the scramble operation of the film, as the program proceeds on a specific number of frames advanced or reversed. The Code Release button should be released when the sequence check is finished.

The next check is the coding check. Again, only the A button is used. When the page has an A through H answer, the A button will work and the code is correct. On a RETURN page, the A button should not work. If it does, the page is coded incorrectly. If the page is coded as it should be, the machine will not show the next frame until the Code Release button is pressed. If the page is on an I track, neither the A button nor the R button should work if the code is correct, until the Code Release button is pressed.

While the sequence check and the coding check are being performed, the film should also be checked for marks, scratches, general quality, and readability. It may be possible to correct such discrepancies by retouching the negative, or it may be necessary to have the film re-shot. If everything checks out as it should, the negative may be released for distribution prints.

PROGRAM VALIDATION

The term "validation" is used with reference to programs in preference to "testing" or "evaluation" because it is broader. Although tests are used in validation and an evaluation is a part of the process, our fundamental interest is in knowing how the student's behavior differs after completion of the programmed course of study.

Although the validation process should be a continuing one, there are two main points of focus. The first is concerned with the early developmental stages of the manuscript and involves a very limited number of typical students. This process has been discussed under the heading Pre-Production Evaluation. The second point of focus centers on the program after it is developed in its preliminary form. Beyond this, validation is concerned with revisions of the initial program.

Validation Test Designs

The first step in field testing is to establish the objectives or purpose for the testing. The simplest form of field testing has as its purpose to measure the amount of learning that takes place as a result of the study of the program. In this case, the design or plan might merely consist of administering a pre-test covering the information contained in the program prior to the students' study with the program, and then an equivalent post-test after the students had completed the program. The difference in scores, if positive, would be interpreted as the learning gain.

This design ignores many factors that are frequently important in the validation of a program. For instance, the students may actually learn a great deal simply from taking the pre-test. Also, it may be that the students do learn from the program but could learn as efficiently or as quickly by some other means. This would not be shown in such a simple test design.

The next simplest design consists of administering the pre-test to two matched groups of students. One group is then given the program and the second group is occupied in other ways for an equal length of time. Then, the post-test is administered to both groups again. This method effectively cancels out any learning that results from the administration of the pre-test. Of course, it does not compare the programmed learning with any other form of instruction.

A third field test design also uses pre- and post-tests with two groups but provides a different form of instruction covering the same material for the second group. A major problem with this design is that the alternative method of instruction is seldom standardized. Thus, it is difficult to generalize the results to other situations.

For instance, a very poor instructor working with the control group would probably cause the test scores of the control group to be much lower than the group taking the programmed instruction. Sometimes an attempt is made to avoid this criticism by providing the live instructor with the programmed learning course to be used as a course outline. This puts the program at a disadvantage because the live instructor can be more flexible and adaptive than even the most elaborate program. Also, this is a somewhat unrealistic situation since the program would not generally be used in this manner in live teaching situations.

These are the major methods of validating programmed instruction and the problems that accompany each. Other less frequently used and specialized validation techniques will be discussed later. Of these principally used methods, it is the project director's responsibility to evaluate the advantages and the problems against the expense of the various designs to determine which technique is most appropriate for the specific situation or use.

Regardless of the design chosen, a pre- and post-test will be required. Construction of these tests is critical and will be examined.

The objectives of the pre- and post-tests should be equivalent and the tests should be so constructed that the content and instructional aim of the program are accurately represented. One way of doing this is through the use of the Criterion Test Analysis form. This is shown as figure 24 and is a variation of a two-way chart described by Victor H. Nowl in his book Introduction to Educational Measurement, published by Houghton Mifflin, Boston, 1957.

The chart provides a means of listing the elements of the program (program content) on the lines of the left-hand side of the chart and the objectives for the program in the column. In many cases, it will be desirable to list objectives specific to the program in reference under the general categories shown on the form.

TOPIC	Knowledge	Skill	Performance	Attitude	Safety	TOTAL

Figure 24

Criterion Test Analysis

Using this form, the project director determines the relative weight of emphasis that is to be placed on each element of the program's content in reference to each of the program's objectives. This is usually stated in terms of the number of questions on the pre- and post-tests that are to be written to evaluate that combination. When this is done, the columns and lines are totaled across and down. The totals should be the same in both directions and represent the number of items required for the pre- and post-test.

Using this information, the program writer constructs a single test with two questions covering each item.

This single test then becomes the basis for both the pre- and post-tests. The equivalent questions for each item are separated on a random basis. One method for doing this is to take the first alternative for the odd-number items and combine them to form one of the two alternate forms. The remaining items complete the other alternate forms. The two forms are arbitrarily designated pre and post.

Test Subjects

It is desirable and usually essential to use some objective criteria for the selection of the students comprising the test groups. The objectives for the program and the assumptions concerning the students will, of course, provide broad guide lines but these do not usually provide the means of screening or selecting. The most frequently used screening devices are standardized objective tests such as intelligence tests, math tests, spelling tests, manual dexterity tests, or others that seem appropriate to the function of the program. This information may or may not be correlated with results of the pre- and post-tests depending on the validation study's objectives.

Mechanical Recording of Validation Data

The easiest and most reliable means of accumulating individual records of performance data is through the use of mechanical recording equipment. Four such systems are available for use with the AutoTutor Mark II. These are described below.

Techni-Riter

Description: The Techni-Riter Model TR 120 is available as a ten or twenty channel inkless event recorder. It makes ten to twenty simultaneous rectilinear recordings on heat sensitive chart paper. This is done by moving the chart paper under heated knife edge writing styluses. The recorder features convenient "drop in" loading of 200 ft. rolls of chart paper and a four-speed electric drive transmission.

When adapted to the AutoTutor Mark II, one stylus is connected to each of the ten AutoTutor operation buttons. A twenty channel Techni-Riter can accomodate two AutoTutor machines simultaneously.

Data Yield: Each button pressed on the AutoTutor is recorded as a deflection of one or more of the styluses on the Techni-Riter. Time is established by interpreting the graphed chart paper distance between events. In some operations of the AutoTutor two or more of the styluses will indicate a simultaneous event. In this case, the highest numbered stylus is used in reading the chart.

DuKane Model No. 99A 230

Description: The DuKane is a paper tape recorder that stores the written responses of the subject inside the machine. The paper tape which is loaded inside the machine runs past a slot in the machine, half of which is covered with glass. The subject writes his answer into the open area and pushes a button on the side of the machine. The tape then advances, carrying the answer under the glass portion, of the slot, preventing him from changing his answer. The machine is immediately ready to record the next response.

When adapted to the AutoTutor Mark II, the recorder is interlocked with the teaching machine so that the following cycle is mandatory:

1. The student reads the frame on the AutoTutor and makes a decision concerning the correct response. At this point the buttons of the AutoTutor have been rendered inoperative. This reminds the student that he must record his response on the DuKane before he can advance to the next image of the AutoTutor.

2. After recording his response on the DuKane recorder, he presses the advance button which moves the tape under the glass window, preventing further change. This action renders the DuKane inoperative and energizes the AutoTutor. Thus, the student is required to indicate his response to the AutoTutor and complete the cycle before indicating another response on the DuKane recorder.

3. Pressing a response button on the AutoTutor again neutralizes the transport mechanism of the teaching machine and energizes the DuKane recorder mechanism, completing the cycle.

Data Yield: The DuKane paper tape recorder yields a student response of the sequence of buttons pressed or the sequence of the image numbers as they were viewed. These can be readily compared to a diagram of the program to determine the frames on which errors were made by the student, and to chart his progress through the program. Additional information, such as start and stop time, can also be recorded.

Camera Recorder

Description: The Camera Recorder is a bracket accomodating a movie camera equipped for single frame photography and coupled electrically to the AutoTutor II. The camera is focused to record each image displayed on the AutoTutor. Time or other identifying data can be obtained by including it in the area of focus.

Data Yield: This recorder yields a photographic record of the frames viewed in the sequential order of their presentation. Time data is available by mounting a clock or timer within the photographed area.

Baronoff Recorder

Description: The Baronoff is a printed paper tape recorder that adds, subtracts and prints the total impulses received at the end of each cycle. Incorporated in the circuit is a "reset to zero" switch and a "slew" switch which permits setting the counter-printer to any predetermined digit combination. These features permit synchronization with the frame numbers on the AutoTutor.

When adapted to the AutoTutor Mark II, the following sequence of events occurs:

1. The AutoTutor is loaded and positioned on the zero frame of the program.

2. The Baronoff Printer is loaded and its counter reset to zero.

3. Each time the AutoTutor transport mechanism operates, an electrical impulse is fed to the Baronoff Printer for each frame advanced. These impulses are summed and the total, which corresponds to the new image number displayed, is printed on the Baronoff tape.

4. When the AutoTutor transport mechanism moves in reverse, the electrical impulses to the recorder cause it to subtract one digit for each impulse and print this new total on the tape.

Data Yield: The tape of the Baronoff recorder yields a printed record of the numbers of each image viewed and the order of their presentation. This record can be compared to a diagram of the program to determine the frames on which errors were made by the student and chart his progress. The Baronoff Recorder does not yield time information.

The data obtained from many of these devices can be used for data input in any of the validation techniques which we have described. Of course, many of the techniques involve additional information which must be obtained from other sources.

Analysis of Validation Data

After the field trial has been completed, the accumulated data is tabulated and analyzed. The most frequently used statistical procedures are group mean and standard deviation based on the group size. These statistics readily lend themselves to familiar tests of significance.

Unfortunately, most validation studies of programmed instruction at the present time are limited to the field testing techniques described in the previous section. Many times additional data that is available is ignored. For instance, data from standardized tests used for the selection of the student population, are frequently ignored in the statistical analysis of the pre- and post-test data. The mean and standard deviation show only the average and the range for the tested groups. They do not show the individual differences that exist within the group.

One standard method of accomplishing this, where selection data is available, is to correlate the selection data with the pre- and post-test data. But shortcut methods are available for use in determining fruitful avenues for a more comprehensive analysis.

For instance, a simple method of establishing a pattern of individual differences in relationship of two factors involves the use of a chi-square table. Here, one factor such as intelligence or verbal facility can be related on an individual basis to a second factor such as the pre- and post-test score difference.

A chi-square table of this type will indicate whether or not the pattern of individual differences exists in relationship to the variables. It will also indicate whether the pattern is negative or positive. From this can be determined the statistical degree of confidence of the pattern of differences. A trend indicated by this method of analysis can be followed up with more precise techniques.

Interpretation of Validation Data

At this point, it is possible for the project director to determine whether the program actually accomplishes its mission. This evaluation is, of course, useful in determining that the student is able to participate successfully in the next unit of study. It also makes possible the appraisal of the unit selection criteria, the assumptions concerning the students, the choice of the program paradigm, and the degree of accomplishment of the program's objectives.

If the evaluation procedure indicates that the program does not accomplish the program's objectives, additional editing and revision are needed.

At this point, the project director would re-examine each of the preparatory steps in relation to the results in an attempt to uncover the factors responsible. The following questions may be pertinent: Did the program's length preclude the proper development of facts, concepts and skills? Did the established level of difficulty or length of the program eliminate sufficient repetition or drill of significant concepts? Were the assumptions concerning the characteristics of the students valid? Does the program build the proper basis for those terminal behaviors previously selected? Did the program include a sufficient sample of behaviors selected as appropriate? Was there adequate provision for practice in discrimination between appropriate and inappropriate behavior?

Example of a Special Validation Design

Now we will examine a validation study that employs five separate study designs. The overall purpose of the study is to evaluate the effectiveness of programmed learning, in salvaging students that are failing a training course in electronics. This is an attempt to significantly reduce the washback and drop-out rate. In broad general terms, the study is designed to evaluate the effect on marginal students of pre-study, post-study, study hall techniques and programmed automatic instruction using the AutoTutor. All the students in all study designs take the first two weeks of an electronics course taught by live instructors.

Study A is designed to measure the effect of pre-study of the third week material by the AutoTutor teaching machine on the students' achievement during the third week. At the end of the second week, an examination is administered. All the students who score below 60 must repeat the second week. Students who score over 75 are allowed to advance to the third week and will randomly be divided into the experimental and control groups. The control group is allowed to pass on to the third week without remedial instruction. The experimental group will spend their two-day weekend on the AutoTutor studying the third week material in advance and will then join the normal advance control groups at the start of the third-week. At the end of the third-week, the experimental and control groups will be compared on the third-week examinations. A diagram of this design is shown as figure 25.

The second design we will designate is Study B. Its purpose is to assess the effectiveness of programmed learning in salvaging the students failing to attain a passing score on the third-week examination. Using the third-week examination scores, the class will be divided into three groups; a normal advance group with scores of 75 or better; a marginal group with scores between 60 and 75; and a washback group with scores below 60. The washback group will repeat the third week or be dropped from the course at the discretion of the director. The normal advance group will proceed to the fourth week as scheduled. The marginal group with scores of 60–75 will be randomly divided into experimental and control groups. The experimental group will receive two days of programmed learning via the AutoTutor. On Sunday, both the experimental and control groups will be examined with alternate forms of the third-week examination. Students failing the alternate examination will washback; successful performance will result in a normal advance to the fourth week. A comparative evaluation of achievement gains obtained by the experimental and control groups will be carried out. The design for this study is shown as figure 26. The third design will evaluate the relative effectiveness of compulsory study hall as opposed to programmed learning with marginal students. This will be designated as Study C.

At the conclusion of the second week, those students with scores of 75 or better will advance to the third week as normally scheduled. The students with scores of 70 or less will washback to the beginning for a week for remedial instruction. Those students having scores of 70 to 75 will be divided into the experimental and control groups. The control group will receive compulsory study hall while the experimental group will receive programmed learning on the AutoTutor. A comparative evaluation of the experimental and control groups will be carried out using the third-week examination performance. This study design is shown as figure 27.

The fourth design, which we will call Study D, is an evaluation of pre-entrance exposure to programmed learning materials using typical students. For this purpose, a class formed of personnel awaiting electronics instruction will be divided into two matched groups. One of the groups (experimental) will be assigned three weeks (half days) study of the programmed electronics course on the AutoTutor. The other group (control) will be assigned to routine work details. The 60 hours of programmed material parallels the electronic instruction of the second, third and fourth weeks in the regular course.

After a one week break, the class consisting of 16 program exposed and 16 uninstructed students will enter the live instructor electronics program. At each of the examination points after the second week, the program exposed (experimental) and the uninstructed (control) groups will be compared. The design for this phase of the study will be shown as figure 28.

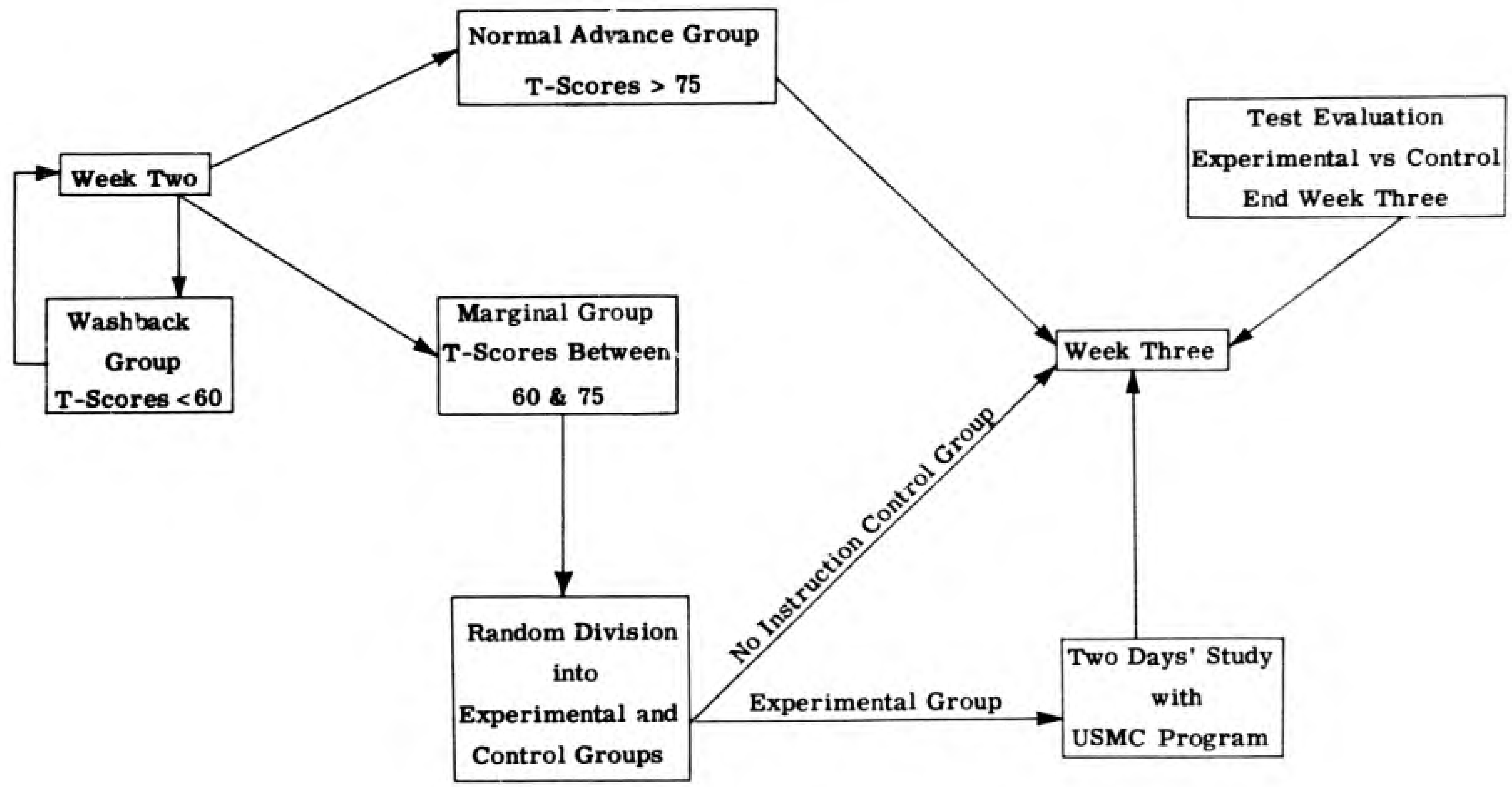


Figure 25
Study A - Pre-Study Application

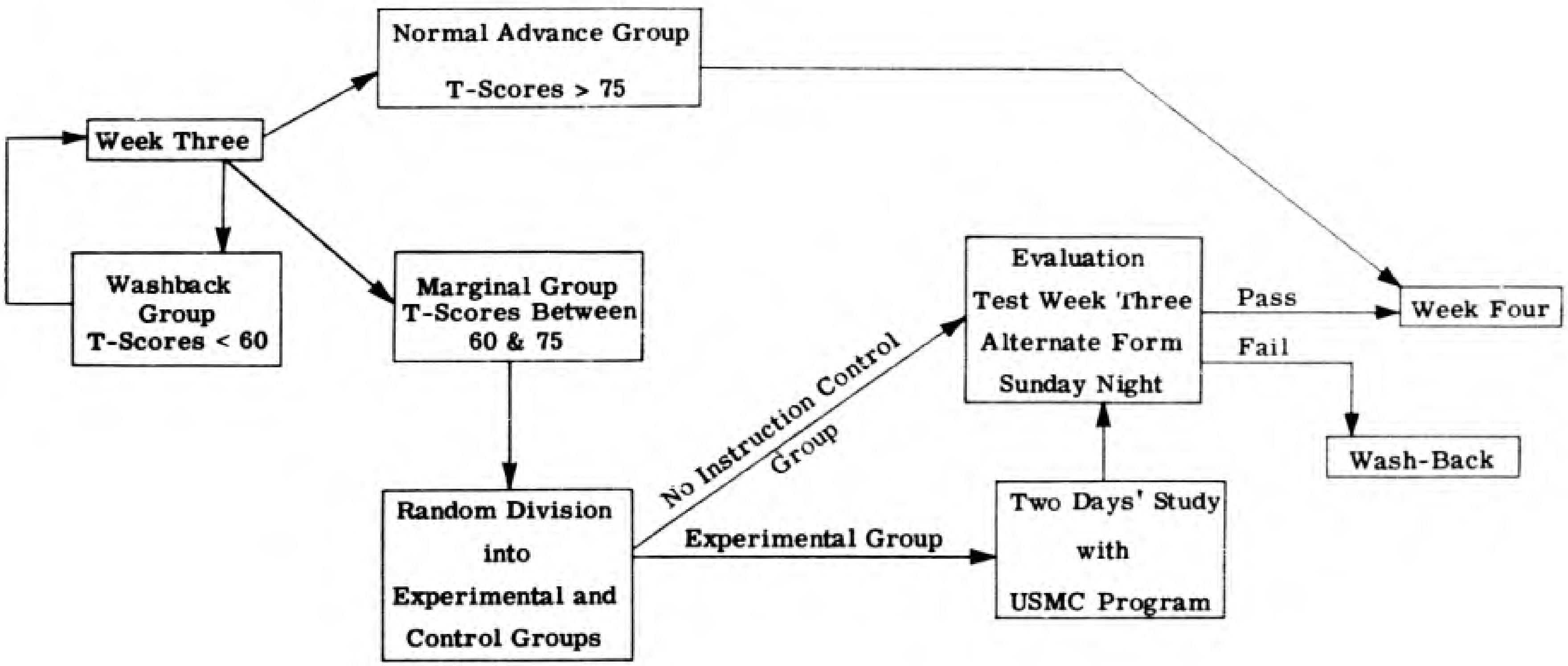


Figure 26
Study B - Post-Study Application

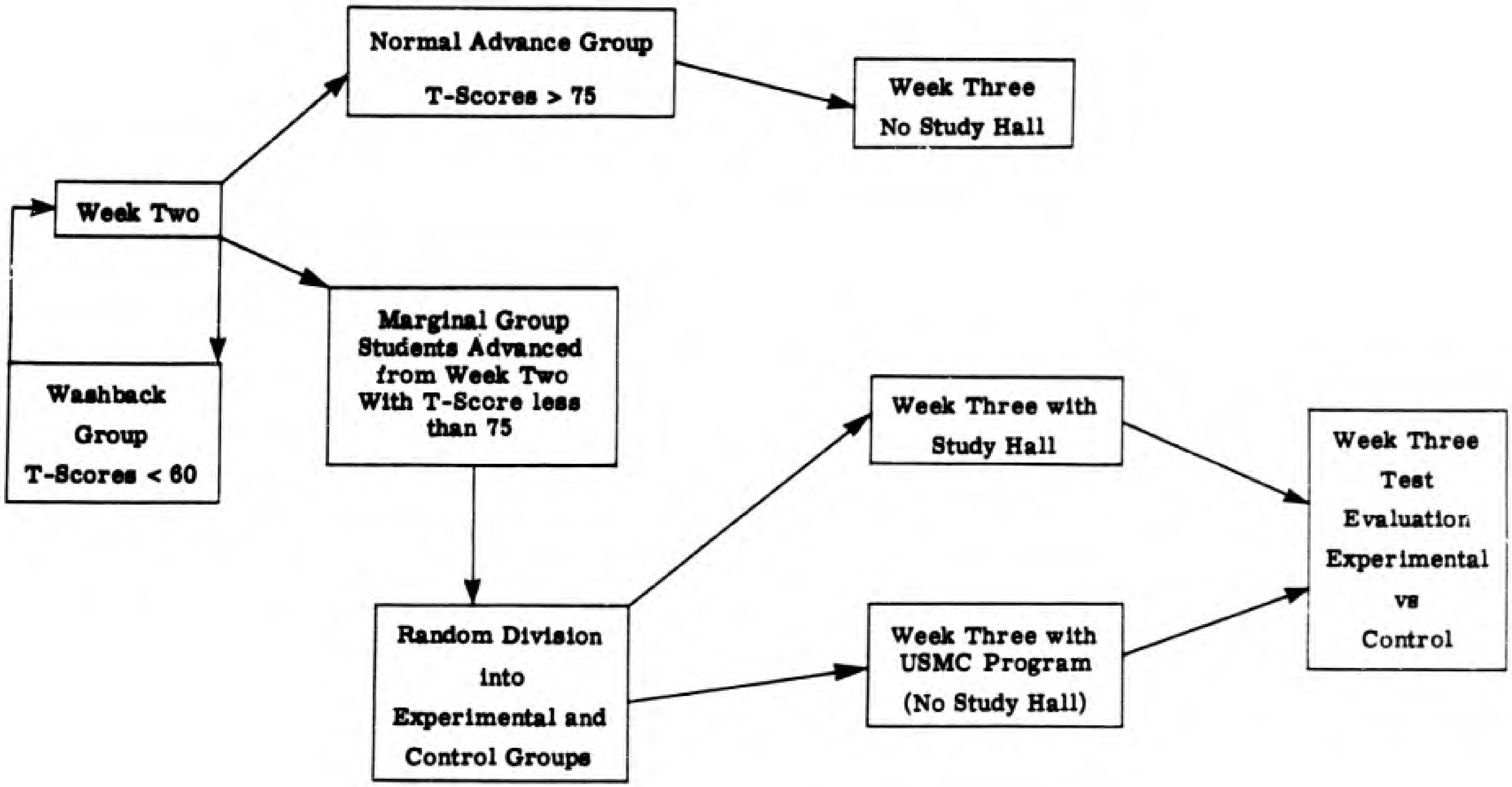


Figure 27
Study C - Study Hall Application

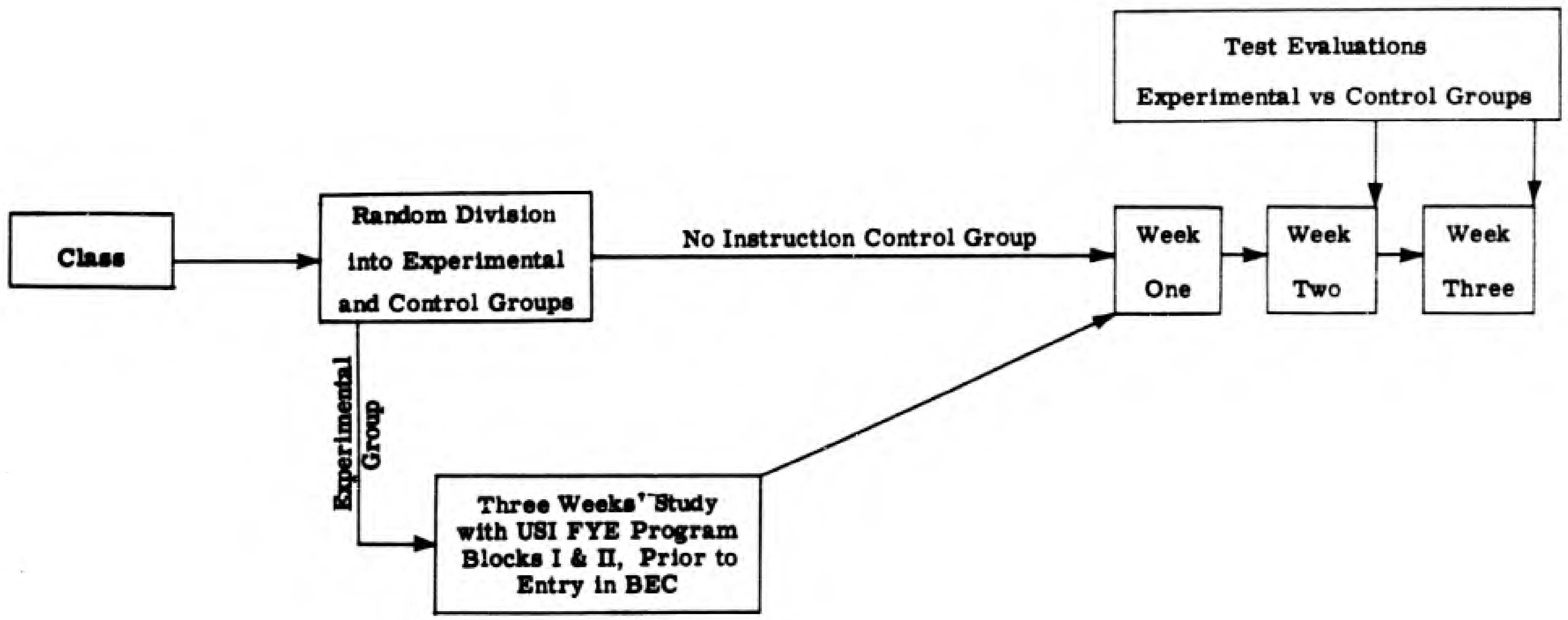


Figure 28

Study D - "Typical" Student Pre-Entrance Application

The fifth and last design will be designated as Study E. It is intended to measure the effect of pre-entrance exposure to programmed learning on marginal students. Two pre-selected classes, arbitrarily designated classes A and B will be divided into upper and lower halves based on AACT scores. The lower half of Class A will be the experimental group, while the lower half of Class B will be the (no instruction) control group. The experimental group will study program material covering the second and third weeks on the AutoTutor prior to entry into the regular live instruction class. Comparative evaluation associated with the achievements of the experimental and control groups in the live instruction class will be carried out. Normally washback and drop-out procedure will be used. This design is shown as figure 29.

The examples here by no means exhaust the special techniques and designs that are useful in validating programmed learning. However, they do give a reasonably broad sample that may suggest approaches to individual special problems. The statistical procedures used in programmed learning validation are the same as those used in other educational and pyschological experiments. The same general precautions should be followed in selecting the statistical process employed.

General Program Evaluation

A suggested checklist for program evaluation is shown as figure 30. Much of the data comprising this list is not, at present, published by the producers of programmed learning material. All of it is or should be available from the producers of these materials upon request.

Provision is made to the right of each item for recording the specific data in reference to the program under consideration. The two columns on the left are provided for individual evaluation of the adequacy of the program in reference to the specified data.

A program may be satisfactory in one application but unsatisfactory for another. This evaluation must be based on the anticipated use of the material.

The date of the last revision is more important in areas of rapid technical change or advance than in basic areas such as grammar and fundamental math. The outline should, of course, conform to the requirements of the user. The statement of the course objectives should be specific and measurable. The performance should indicate competency in achieving the stated objectives. The required study time should be reasonable for the amount of material which the program covers.

Very few programs will be rated completely satisfactory on all criteria unless the user's intended use is identical to the producer's intended application. The degree to which these elements can differ will depend largely upon the nature of the user's intended application. As more experience is gained in the use and preparation of programmed materials, the techniques for evaluation and selection will undoubtedly become more refined. However, even at the present state of the art, programmed learning is indeed an effective training technique.

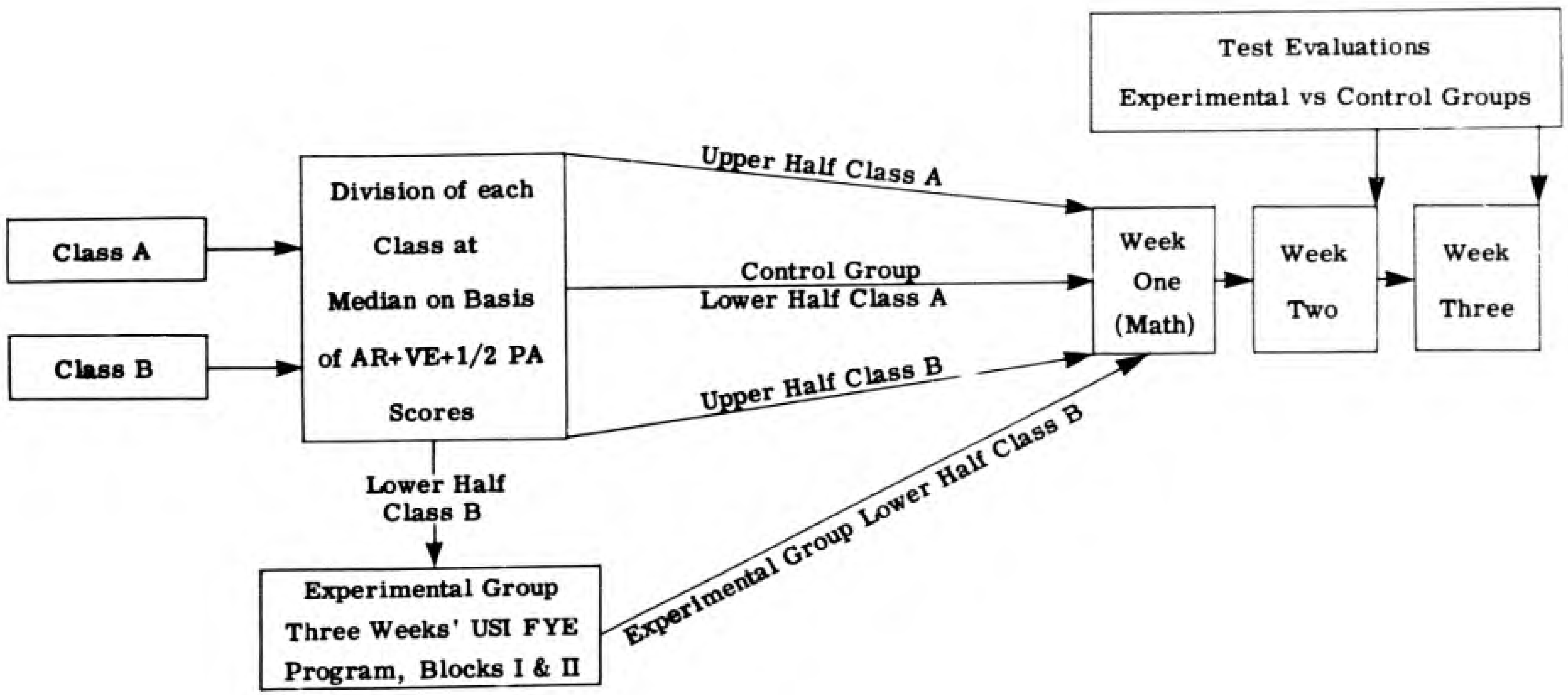


Figure 29

Study E - "Special" Student Pre-Entrance Study Application

Program Title ______________________

Producer ______________________

Satisfactory	Unsatisfactory	Item	Data
		Last revision or copyright date	
		Outline of the program content	
		Statement of course objectives	
		Copy of the performance test	
		Description of validation population	
		Number	
		Achievement Levels	
		Educational Levels	
		Reading Level	
		Performance Scores	
		Administration or study time	
		Number of frames or pages	
		Format—intrinsic, linear, mathetics, etc.	
		Workbook or other materials required	
		Authors and associates	
		Instructor's guide	
		Recommendations for use	
		Price	

Figure 30

Check List for Program Evaluation

Appendix A

IMPLEMENTATION AND UTILIZATION OF PROGRAMS

Programmed learning has proved its usefulness in every phase of education; yet there is still a great deal of controversy concerning how it should be implemented. Part of this controversy stems from the fact that programmed learning usually affects the internal arrangement of the instructional procedure. The group or class is the basic unit for most training and instruction, and for administrative convenience the tolerance for individual differences from the group mean is generally limited. Programmed learning, on the other hand, is individually administered. The self-pacing feature of programmed learning capitalizes upon individual differences. The administrative limits on individual variation that previously held the group or class together as a manageable unit are destroyed by the program. The full range of individual differences exposes the true heterogeneous nature of the group and makes the traditional formulas for group administration unworkable.

In an attempt to minimize the administrative problems that arise, the trend has been to employ programmed learning to complement the other and more traditional form of group instruction. This has been done in a variety of ways. Sometimes a programmed unit is used to replace a given area of the customary form of instruction. This may be a basic fundamental of the course or specific topics within the course that have proven to be difficult topics when covered by other methods. In this application, the entire group is usually given the program at one time. The theory beyond this approach is that when using short units of programmed instruction, the group does not have an opportunity to "spread out" very much. In other instances, short programs have been used to cover pre-requisite material or to provide remedial material for the student who is unable to keep up with the group or class. In these cases, the program is actually used to minimize already existing individual differences within the group. A third approach to integrating programmed instruction into an existing group-oriented administration is through its use as enrichment material. Here short units related to the regular curriculum are provided to selected students to enhance their understanding of their regular work. In this case, the program is frequently used in the manner that "busy work" was formally used to provide the teacher an opportunity to give more attention to the slower members of the class. Enrichment materials sometimes take the form of advanced subject matter and at other times provide problem-solving opportunities in the context of the regular curriculum.

A fourth method of integrating programmed learning material into a conventional instructional situation is through its use as a medium for review. Programmed learning reviews may cover portions of previous instruction or may be used as a standardized form of review at the end of a course. In either case, programmed learning seems to work very well.

The instances just described are typical of situations where the use of programmed learning has been superimposed upon an instructional procedure that is already in existence. Programmed learning's greatest successes to date have been in areas where conventional training and instructional programs do not exist, because of inherent situational problems.

One such problem is geographic or temporal distribution of the trainees. In situations where a number of individuals who require training in a specific function or skill are widely distributed geographically, there is no other method of training that can approach programmed learning in effectiveness and economy. This also applies to situations where the individuals may be centrally located but, because of their work responsibilities, they cannot be assembled into a training group. A similar situation exists in organizations where individuals requiring specific training are brought into the organization in very small numbers. Another area where programmed learning fulfills an unmet need is in situations where other forms of instruction are not available. Small organizations and units that could not support a live instructor are able to provide themselves with the highest level of training competence through the use of programmed materials. This serves to increase the organizational unit's flexibility in the training that it can provide. Whereas a live instructor is limited in his areas of proficiency, the teaching machine affords useful instruction on whatever subject is desired, providing a program on that topic is available.

Because of the novelty of programmed learning and the administrative problems that frequently accompany its implementation, careful planning must be provided to ensure maximum effectiveness. Much publicity concerning programmed learning has led to unwarranted expectations both positive and negative.

Students initially welcome the introduction of programmed learning on the assumption that it is "automatic instruction" that requires no effort on their part. Unfortunately, the learning process is such that a definite relationship exists between student involvement and energy expended and the effectiveness of the instruction. This is less of a problem with intrinsically programmed materials than with other forms of programmed instruction because the psychological principles on which intrinsic programming is based tend to promote student involvement and motivate the problem-solving process.

Training instructors may be apprehensive over the implementation of programmed instruction for a number of reasons. Those who feel inadequate may feel that their inadequacies will be revealed through improved student performance. Even fully competent instructors may fear that extensive use of programmed materials will eliminate their positions or reduce their opportunities.

From an administrative standpoint, instructors and administrators may resist programmed learning because of the problems it presents in scoring and evaluating. The traditional grading system is based on the assumption that some students in any group will learn more or perform more effectively than others. Programmed learning minimizes these differences. A group that is very heterogeneous in reference to performance on stated criteria at the beginning of a course becomes very homogeneous after the completion of programmed learning designed to provide skill measurable by those criteria.

These problems coupled with the ones discussed under the "Logistics of Programming" must be faced by those responsible for the decision to implement programmed learning. Most of these problems can be solved by careful laying of the groundwork after the decision has been made and before the new materials are introduced into the classroom.

The most effective way to overcome the obstacle of acceptance is through personal involvement of those who will be connected with the training. This involvement should take place as early in the planning stages as is feasible. The anticipated problems should be discussed and the aid of those affected should be enlisted in proposing solutions.

It is difficult to provide general solutions that will apply to a wide variety of specific installations. If a long course is to be taught by programmed learning to a large group, then a major problem will be the administrative handling of the wide individual differences in the rate of completion. If short individual remedial units are used, then the problems will not be concerned with the differences in the rate of completion, but rather will probably focus on individual schedules on the machines or on keeping special records of performance.

In most cases, it is wise to enlist the services of experienced and qualified organizations in planning and implementing a major installation involving programmed learning materials.

Appendix B

PROOFREADING SYMBOLS

The following symbols are suggested for use in editing and correcting copy;

1) Cover up all erroneous words with a [redacted] pencil line.

2) Don't use "strike-overs," like this example. Retype, erase, or write-in, so that the word is plain. The strike-over is a potent source of error.

3) If you write a word (or phrase) above a crossed-out word or phrase, indicate it thus.

4) If you insert (a word or phrase) in a sentence, do it thus.

5) If you jump ~~now is the time for all good men to come to the~~ from ~~aid of their country~~ one line to ~~around the bend and through the trees~~ another, draw a "map" thus.

6) Suppose you want to start a new paragraph here. ¶ This is the mark which signifies your intention.

   But if you do not want to start a paragraph here, add a line.

7) Circle all instructions to typist. Circled words should not be typed.

8) To indent material, use this mark.

9) To center material, do this.

10) To place material flush with margin, do this.

11) Set this in *italics*. it

12) Set this **boldface**. bf

13) Underline this.

14) When you have scratched out a STET ~~word~~ and want to put it back in the copy write "stet" (it means, "Let it stand") in the margin or directly above the words to be retained.

15) ~~This is the mark for capitalizing a letter~~ to cross out a whole ¶

16) This is the mark for capitalizing a letter such as barbara.

17) This is the mark For making a Capital letter lower-case.

18) Commas are inserted in copy with this mark, if you really want a comma.

19) Set off "quotation marks" clearly when written in.

20) Set this WORD in capitals.

21) Periods are written like this⊙

22) To eliminate a single letter, do it thisa way or like thiis.

23) To close up a spa ce, do it thus.

24) To insert a space, use thissign.

25) To change a single letter or two because your tepewritfer slipped, do it thus.

26) four means "use figures," like this: 4.

27) 0° means "spell out," like this: zero degrees.

28) To transpose a pair of words, simply draw kind this of a line. To transpose letters, od this.

29) To pick up a and put it elsewhere, word do this.

30) Be neat. If it's a mess, do it over.

32) To move down, make this mark.

31) To move a whole paragraph to where it belongs, draw a clear and bold line.

33) To move type up, make this mark.

Appendix C

GLOSSARY OF GENERAL TERMS USED IN PROGRAMMED INSTRUCTION

This list was prepared under the direction of Dr. Robert Orlando of the University of Minnesota.

ABSTRACTION: A response to a single property of a complex stimulus. Example: Response "Carnivore" to picture of a German Shepherd.

ACCESS TIME: Amount of time elapsed in getting to an item in a program. In computer terminology this is the amount of time it takes the machine to search for a particular bit of information and produce it.

ADAPTIVE TEACHING MACHINE: Refers to teaching machines which automatically alter the instructional presentation sequence as a function of the pupil's performance. Example: The machine may shift to a smaller step size if the pupil is making more than four incorrect responses out of every ten frames.

ADAPTIVITY: The capacity of the teaching machine and its associated program to adjust in one or more ways, on the basis of the learner's responses, to his specific needs.

ADJUSTED-LEARNING METHOD - (See: Branching)

ASYNCHRONY: A method of presenting items in which (a) a changing stimulus requires the same responses or (b) a fixed stimulus requires changing responses.

Example of (a):	$5 + 3 = 8$	Example of (b):	$8 = 6 + 2$
	$6 + 2 = 8$		$8 = 7 + 1$
	$7 + 1 = 8$		$8 = 5 + 3$

AUGMENTING: A programming technique which builds a new concept or complex set of relationships by serially introducing bits of information in such a way that each is related to the other. Example: "John is not a woman." "John is a MAN." "John will die." "John is a person." "People who die are mortal." "John is a mortal." "John is a MORTAL MAN."

AutoTutor Mark II: A brand name (Reg. U. S. Pat. Off.) for a teaching machine produced and leased by U. S. Industries, Inc.

BINARY FRAME SELECTION: The simplest form of branching. There are at most only two alternative routes at any point in the program.

BRANCH: Any one of the alternative routes in a branching or intrinsic program.

BRANCHING: A style of programming in which selection of the next frame to be presented depends on the response given in the current frame. Example I: If an incorrect response is given, the following frame will consist of material which explains in detail why the given response is not correct, whereas if the original response is correct, the next frame presented consists of the next item in the sequence of subject matter. Example II: In a program designed to handle neurotic fears, the question "Which are you more afraid of, dogs, bears, or snakes?" could appear on one frame. Depending on the answer given (dogs, bears, or snakes), the program in the next frame would direct the user to one of three sequences.

CARTESIAN METHOD: An approach to handling complex problems which may be useful in programming subject matter. It involves essentially two activities: (a) analyzing the problem into its smallest parts, and (b) proceeding from the simple to the complex.

CHOICES: Refers to the selection of an answer from several alternatives presented to the student, as opposed to having the student construct or write out an answer.

CLASS-DESCRIPTIVE METHOD: A method of sequencing material in a program, in which the common and significant characteristics of a set of objects are simultaneously presented to the learner. (This is in contrast to the object-desc iptive method.) Example: The heads of three different coins are presented side by side as one display - the tails of the coins as another display.

COLLATOR: Component of a teaching machine which measures and records the learning process by collecting and recording data such as the number of errors, the type of error, time intervals required for response, etc., in such a way that each item is collated with the part of the program to which it pertains.

COLLATOR-RECORDER: The component of a teaching machine which records number of errors, type of error, response latency, etc., by frame number.

COMPARATOR: Component of a teaching machine which judges the "correctness" of the pupil's response. This evaluation is then transmitted, depending on the mode of operation, to the pupil, the reinforcement dispenser, the collator, and/or the sequence control unit.

CONFIRMATION SEQUENCE: A sequence in which the learner (a) sees the cue-stimulus, (b) makes the response, and (c) sees the confirmation. This is typical of a memory drum.

CONSTRUCTED RESPONSE. A response which is chosen from a virtually unlimited number of alternatives based on the student's past experience. The response, however, may be limited to a general type, such as words or numbers. The constructed response is contrasted with a response to a multiple-choice question.

CONSTRUCTION: The process of requiring the subject to write out or prepare an answer as opposed to choosing one of several alternative answers.

COPYING FRAME: A type of frame which uses the "copying" programming technique to elicit the correct response under appropriate conditions. The pupil merely duplicates the response portion of the frame, which is given in full.

CORRECTION PROCEDURE: Under correction procedure a pupil is required to make a correct response to a frame before he is permitted to continue.

CRITERION PROGRAMMING: The subject matter is not programmed; instead, programmed instructions direct the student to resource materials from which he returns for a test of the material (criterion test) and his next instructions.

CUE STIMULUS: The material (problem, statement) which precedes the blank in a frame. The task of the learner is to make the correct response to the cue-stimulus.

CUMULATIVE DENSITY: The density of any portion of a program in its relation to materials which have been presented in preceding lessons.

CUTBACK-PAGE BOOKLET: A booklet designed with each succeeding page narrower than the preceding one. An answer sheet is placed under the smallest page so that its right edge extends beyond the rest of the text, providing blanks for responding to frames in the text. The correct answers are printed on the answer sheet in a position such that the frames to which the answers belong are on a page above the answer sheet which must be turned to expose the answers.

DENSITY: The rate at which new material is introduced. It is the ratio of the number of different responses required of the learner to the total number of responses required. Density is independent for a single set of frames and cumulative for a sequence of sets. If all the responses are different, the density is 1.00.

DIFFERENTIAL PROGRAMMING: In differential programming, variation in the program is a function of pupil characteristics.

DISPLAY: Presentation of subject-matter information to the learner by a teaching machine. May utilize visual, auditory, or tactile communication channels, or combinations of these.

DISPLAY MECHANISM: The unit of a teaching machine which presents the content material in a series of frames.

DOVETAILING: A technique used in programming which involves the interlacing of associations into a pattern consistent with that required when the information concept or skill is put to use. Example: "If $5 \times 5 = 5^2$, and $7 \times 7 \times 7 = 7^3$, $N \times N =$ _____ ?"

DROP-OUT: Feature of a teaching machine constructed so that when the student responds appropriately to a given frame, that frame is omitted on subsequent repetition of the program.

ERROR: An incorrect or inappropriate answer to a specific item in the program.

ERROR RATE: Refers to the number or percentage of a given group of subjects incorrectly responding to a specific item on the program. A high degree of error would probably indicate a need for revision in the program.

EXTRINSIC PROGRAM: Synonymous with the term "linear program."

FADING: A technique used in programming: the gradual withdrawal of stimulus support. Fading can be done in intact units (all materials to be faded can be dropped out at the same time) or portions can be faded progressively. This technique lends itself especially to material where visual discriminations are important; for example, map reading.

FEEDBACK: The function of a teaching machine which consists of providing the pupil with "knowledge of results."

FORCING: The presentation of subject matter in such small steps as to assure a correct response by the learner.

FRAME: A unit of a program; the segment displayed at each step in the sequence. Usually the unit that requires a response.

FRAME, FORCED: A stimulus frame presented to the student forcing him to respond correctly by making the answer obvious.

FRAME, TERMINAL: A frame having no prompts, located at the end of a sequence, designed to test whether the student has reached "terminal behavior."

HARDWARE: Slang term referring to mechanical teaching devices.

HINTS: Devices used to direct the student's behavior in the desired direction. Used to increase the likelihood of a correct response.

ITEM: Any single unit of a test or experiment - e.g., a single question on a test or a single nonsense syllable in a list of syllables.

ITEM, AUGMENTING: An item supplying new information but not requiring the student to make a relevant response.

ITEM, DELAYED REVIEW: An item which allows for the distribution of practice. Differs from other items only in terms of presentation.

ITEM, DOVETAILING: An item requiring the student to make separate responses to separate stimuli which otherwise may become confused.

ITEM, FADING: An item requiring the student to review what has been presented. In addition the item withdraws information successively. Similar to Skinner's "vanishing technique."

ITEM, GENERALIZED: Items which summarize the common characteristics of several specific items already presented to the student; e.g., after learning in previous items that "I, you, he," are pronouns, he is given the problem "all words which are used in place of a noun are called ________."

ITEM, INTERLOCKING: An item that requires a student to review the established skills while new information is being presented.

ITEM, LEAD IN: An item not requiring new information or rehearsal of the skill where a problem is restated.

ITEM, RESTATED REVIEW: An item requiring a rehearsal of the skill where a problem is restated.

ITEM, ROTE REVIEW: An item presenting a problem identical to one presented earlier.

ITEM, SPECIFYING: An item which exemplifies a general rule or principle.

ITEM, SUBJECT MATTER: An item classified with respect to its subject matter content.

ITEM, TAB: A specialized term referring to the technique of having the subject pull a tab to indicate his response, instead of writing out an answer or selecting a choice.

LEADING: The student is first asked to talk about familiar things using his everday vocabulary. He is then led to discuss relations among these. Technical terms are then slowly inserted.

LIBRARY UNIT: A component in a teaching machine used to store the program. The selector unit picks items to be presented from the library unit.

LINEAR PROGRAM: A program which has a single, predetermined sequence of steps. Error responses are not corrected or immediately repeated. A drop-out device can be incorporated into the machine whereby errors may be reviewed at the end of the set sequence.

MATCHING: Procedure used in some Skinner machines to inform student of correctness of his response. After writing his response, the student moves a lever which exposes the correct answer with which the student compares or "matches" his response.

MATHETICS: An approach to programming designed by T. F. Gilbert. It is an attempt to establish a set of rules or guidelines for analyzing and writing programs.

MIN-MAX: (minimum complexity... maximum function). The trade name of a teaching machine distributed by Grolier, Inc.

OBJECT-DESCRIPTIVE METHOD: A method of sequencing material in a program in which a complete object is presented and the different components of interest are pointed out or identified. (This is in contrast to the class-descriptive method.) Example: The word "dime" and the tail, head, and side view of a dime are presented in one display.

OBJECTIVE: An objectively defined goal toward which instruction is directed. The pupil is expected to reach this goal by the end of the instructional unit. Example: A "driver education" course may have as an objective: To qualify for a driver's permit by passing, with a score of at least 70%, a true-false and multiple-choice test on road signs and rules.

OVERPROMPTING: Undesirable feature of some programs in which a text of frames has an excessive number of prompts. The student is likely to become overly "dependent" on program-supplied responses, making weaning more difficult.

PACE: The rate at which the subject is permitted to work through the programmed material. (The pace may be determined by the learner or by a pacer.)

PACE, PACING: Time intervals in instruction. Two crucial intervals are: (a) between presentation of a cue or question and presentation of the correct information or answer, and (b) between one cue or question and the next. (See: Pacing, self.)

PACED-PRACTICE MODE: A mode of operation of a teaching machine in which a timer limits the time durations. Example: The time of presentation is limited, or the time for a response is limited.

PACER: 1. Component of a teaching machine which limits the time intervals (a) between presentation of a cue or question and the presentation of the correct information or answer and/or (b) between one cue or question and the next. 2. A type of stimulus-response device which presents stimulus materials for a given interval of time and then provides the appropriate response, whether or not the learner has attempted to answer. Example: Memory drum.

PACING, CONTROLLED RATE: Control of the subject's rate of responding by features of the mechanical device utilized to present the program.

PACING, SELF: The rate at which the subject might complete the material at his own rate depending upon success on the previous steps.

PANEL: A short passage of prose material, graphs, and similar material which is presented or studied along with the discs in the Skinner device.

PANEL (EXHIBIT): Supplementary display of information to be referred to by the pupil as he responds to a set of frames.

PRACTICE MODE: A mode of operation of a teaching machine in which the student continues to make responses to a frame until he makes the correct response. After each response the student is told whether he is correct or not. (See also: Correction Procedure)

PRESSEY DEVICE: The earliest known device (1926) originally developed for use with multiple-choice tests. Device could be set so that items missed could be skipped or repeated until success was established, a raw score obtained, and an item analysis or error count secured.

PREVERBAL MACHINE: A machine which presents frames in the form of pictures or figures which may be matched or contrasted; no words are used in the frames presented. Such programs can be used with pupils of pre-reading age.

PROGRAM: An arranged set of frames in a given subject.

PROGRAM, EXTRINSIC: A term usually applied to Skinnerian (linear) programs. The program proceeds in small steps from simple levels to complex levels in a predetermined order of frames.

PROGRAMMED BOOK: A special book in which the subject matter to be learned has been arranged into a series of sequential steps leading from familiar concepts to new materials. Differs from a "scrambled text" in that the content is arranged so that the student proceeds directly from one step to the next, or one succeeding page to the next, rather than skipping around. The student generally is asked to construct a response as opposed to making a choice among alternatives.

PROGRAMMED TEXTBOOK: A book requiring the learner to construct responses to the systematically arranged materials. It follows the linear approach (Skinner). Example: Page 1 gives the stimulus and requires a constructed response while page 2 gives the feedback (reinforcement), and presents a new stimulus requiring another constructed response... and on to page 3.

PROGRAMMING: The process of arranging the material to be learned into a series of sequential steps; usually moves the student from a familiar background into a complex and new set of concepts, principles, and understandings.

PROGRAMMING, INTRINSIC: A method of programming materials that directs the erring subject along certain corrective pathways before he is permitted to proceed to the next step in the program. Requires that each step contain multiple-choice answers.

PROMPT: Programming technique designed to insure the desired response to a frame.

PROMPT, EMPHASIS: One of the cues or stimuli employed for insuring correct responses. An emphasis prompt is usually an underlined word or phrase written in capital letters to give it emphasis. Example: (a) PARIS is the capital of France. The capital of France is __________

PROMPT, FORMAL: A prompt which provides the pupil with cues to the appearance of the required response: i.e., the way the response "looks." Example: "The capital of France is P---S." The "P," the number of dashes, and the "S" are formal prompts.

PROMPT, SEQUENCE: One of the cues employed for insuring correct responses. The sequence prompt may be one of two formats: (a) If a pupil reads a text in one item and gives one of the words in the next item, the sequence prompt is a function of what the student had just read or seen. (b) If a student copies a model in one item, and repeats the response without a model in the next item, the sequence prompt is a function of what the student had just written or produced.

PROMPT, THEMATIC: A prompt which depends for effect upon previous associations in the pupil's repertoire. Example: Canis Familiaris is man's best friend. Canis Familiaris is the technical name for the animal commonly called a ______. The phrase "man's best friend" means "dog" to most people. The phrase "man's best friend" is the thematic prompt.

PROMPTING: The method or sequence of providing verbal and symbolic cues to encourage responses. Can be visual, verbal, symbolic, or auditory.

PUNCHBOARD DEVICE: A unit containing rows of holes to use in selecting a multiple choice response. The student punches through a paper with a pencil or stylus. If the response is correct, the pencil goes completely through the paper. If it is incorrect, a backing plate prevents the pencil from going through completely.

PURE-PART METHOD: A method of sequencing programmed instruction in which each part is first learned separately to a criterion of mastery. Subsequently the parts are repeated until the whole has been brought to the criterion already achieved by each part.

"QUICKENING" PROCEDURE: Giving the student knowledge of the results of his response while he is still in the process of making the response. Example: After writing the first one or two letters of a response word, the student would be told he is incorrect.

QUINTAIN: A medieval teaching machine used to train knights. The appropriate response was to strike a shield directly in the center with a lance. If the lance struck off center, the device would deliver feedback by striking the horseman a blow as he rode by.

RECYCLING: A machine function which returns the student to a previous part of the program.

REINFORCEMENT, MECHANISM: Some type of reward for responding correctly to the items in the display. A motivational factor causing the individual to keep working at the set of materials. Sometimes considered as an integral part of the confirming mechanism.

REINFORCERS, PRIMARY: A class of stimuli which will, without any prior history of training, reinforce operant behavior.

RELATIONAL GUIDANCE: A technique used when a principle is to be applied in a variety of contexts, each requiring a different objective response.

RESPONSE MODE: The kind of response the pupil makes while working on a program. Examples: writing the answer, pushing a button, pressing a panel, etc.

ROTE REVIEW: Repetition of a frame presented earlier in the program. This type of presentation is useful mainly where sheer memorization of verbal material is desired. Such items are usually presented out of sequence.

RULEG: A system in programming which consists of a statement—such as a rule, principle or definition—followed by one or more examples. The statement and examples may occur in the same frame or in a series of frames.

SCRAMBLED BOOK: (See: Scrambled Textbook)

SCRAMBLED TEXTBOOK: A programmed textbook arranged according to the branching method of programming. If a correct response is made to a question, the learner is referred to a page in the book which confirms his response and moves him on in the program. If the response is incorrect, he is referred to another page which given him remedial attention. Eventually he is returned to the question that he missed, and he tries again. Since the pages are not taken in order, it is called a "scrambled textbook." Each frame presents a multiple choice question; each of the several answers directs the student to a different page.

SEEDING: Inserting review materials at intervals in a program. These reviews may be either regularly spaced (periodic) or irregularly spaced (aperiodic).

SELECTOR UNIT: The component of a teaching machine which compares the pupil's response with the correct response.

SEQUENCING: Arranging the frames of a program in an order that provides the most efficient situation for learning.

SET: A series of frames in the program dealing with one information unit.

SHAPING: The building of a behavior or set of behaviors through the differential reinforcement of progressively more adequate forms of behavior. (Skinner)

SIMULATOR: A type of stimulus-response device which duplicates the essential characteristics of some complex task and which requires appropriate action from time to time on the part of the learner. Ordinarily, this type of device does not provide immediate feedback. Example: Electronic flight simulator.

SINGLE-ERROR-PERMITTED MODE: A mode of operation of a teaching machine in which the student makes a single response to a frame. Then he is told if the response is correct or not, and if the response is incorrect, he is shown the correct response.

SINGLE-TRY MODE: A mode of operation of a teaching machine in which the pupil is allowed only one response to each frame. The next frame is presented whether the response was correct or not.

SKINNER DEVICE: A mechanical device which presents a set of programmed materials. At each step the subject must construct an answer and evaluate its correctness with a model answer before proceeding further in the program. Generally considered the forerunner of later model "teaching machines."

SKINNER DISC: A round, flat, record-like device which contains a set or series of program materials for the Skinner device. Contains the questions to be answered, and spaces for recording student's answers and the correct response, as well as for making a record of successful or unsuccessful performance.

SKIPPING: Termination of program prior to the final item of a subset and going on the next topic. If the student's response is correct on a key frame, the student is directed to the next set: he "skips" a portion of the program.

SOCRATIC METHOD: A method of instruction which consists of a conversational quiz in which a tutor asks questions, the student replies, and the tutor confirms or denies the answer. If a response is incorrect the tutor leads the student by a series of questions to the correct response.

SPECIFYING ITEMS: Items which exemplify a general rule through specific examples, e.g., after a student has been told that the symbol ÷ means "divided by," he is given a problem: 4 ÷ 2 __________. Response to this item is "4 divided by 2."

STABLE SEQUENCE: Refers to the sequence in a serial-learning task, such as learning the letters of the alphabet. It is stable because the order of stimuli and responses remains fixed and predictable.

STEP: The increment in subject-matter level to be learned with each succeeding item or frame in the program.

STEP SIZE: Average amount of difference between successive frames. A function which is inversely proportional to the number of frames in an instructional unit.

SYNCHRONY: Both stimuli and responses change from item to item in the series, or neither the stimulus nor the response changes from item to item. Example of the latter type is repetition.

TEACHING AIDS: Devices useful in the teaching process which do not assure learning either because they do not necessarily require any action on the part of the learner when he is presented with the subject matter, or they permit him to practice some activity but do not necessarily provide him with subject matter on which to practice.

TERMINAL BEHAVIOR: The behavior a program is designed to produce. (See: Objectives)

TEST MODE: An operational mode of a teaching machine which will provide the display unit, the response unit, and the program, but will not provide feedback to the student. E.G., Pressey's Drum Tutor is a testing device.

TRACK, MULTIPLE: A provision within the programmed material for allowing subjects to pursue alternative sub-divisions of the program in terms of their successes or failures with earlier sections of the sequence. (See: Branching)

TRACK, SINGLE: A common set of programmed materials which all subjects work through, there being no alternative program such as in the multiple-track situation. (See: Linear)

UNSTABLE SEQUENCE: Refers to the arbitrary and unpredictable order between items in a paired-associates learning task such as learning a French-English vocabulary. (See: Stable sequence)

VANISHING: A form of fading in which the prompt is gradually removed by making it progressively less discriminable.

WEANING: Behavioral goal of the fading technique. Training of the pupil to make "independent" responses to stimuli which were previously accompanied by prompts.